U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MESTRADO EM CIÊNCIA DA INFORMAÇÃO

RECOMENDAÇÃO DE PERCURSOS CULTURAIS COM ESCASSEZ DE DADOS

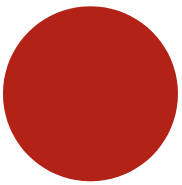
Maria Inês Fontes Rocha

M

2018

UNIDADES ORGÂNICAS ENVOLVIDAS

FACULDADE DE ENGENHARIA
FACULDADE DE LETRAS



Maria Inês Fontes Rocha

Recomendação de Percursos Culturais com Escassez de Dados

Dissertação realizada no âmbito do Mestrado em Ciência da Informação, orientada pelo Professor Doutor João Pedro Mendes Moreira

Faculdade de Engenharia e Faculdade de Letras

Universidade do Porto

Julho de 2018

Recomendação de Percursos Culturais com Escassez de Dados

Maria Inês Fontes Rocha

Dissertação realizada no âmbito do Mestrado em Ciência da Informação, orientada pelo Professor Doutor João Pedro Mendes Moreira

Membros do Júri

Presidente: Professor Doutor António Manuel Lucas Soares

Faculdade de Engenharia - Universidade do Porto

Orientador: Professor Doutor João Pedro Carvalho Leal Mendes Moreira

Faculdade de Engenharia - Universidade do Porto

Arguente: Professor Doutor Rui Miguel Lourenço Lopes

Instituto Superior de Engenharia do Porto (ISEP) - Instituto Politécnico do Porto

„Man muss das Unmögliche versuchen,

um das Mögliche zu erreichen.“

Hermann Hesse

Agradecimentos

Esta tese não teria sido possível sem a inspiração e apoio de várias pessoas:

Gostava de exprimir a minha profunda gratidão ao Professor Doutor João Pedro Mendes Moreira pela supervisão da minha tese. A sua orientação contínua, sugestões e otimismo foram inestimáveis para a elaboração desta tese.

Queria também agradecer à Professora Doutora Manuela Pinto, ao investigador Rodolfo Matos e a Francisca Vasconcelos por me terem apresentado os projetos da MDUP #IWASHERE e #GPSEngenharia que foram o ponto de partida desta tese.

O meu apreço vai ainda para os meus colegas de mestrado que colaboraram em diversos momentos, que tanto me ensinaram e me estimularam sempre que precisei. Nem que fosse pela simples companhia para almoço. Para os meus amigos que me motivaram e ajudaram durante este percurso.

A todos que contribuíram direta ou indiretamente para que esta tese ficasse concluída.

E sobretudo, quero agradecer aos meus pais e ao meu irmão pelo seu apoio emocional e moral nesta etapa singular da minha vida.

Obrigada!

Resumo

Graças às novas tecnologias de informação e comunicação, a partilha do património histórico, cultural e científico não se restringe apenas a uma população fixa num determinado espaço. A Universidade do Porto, com o projeto do Museu Digital da Universidade do Porto, visa partilhar o seu espólio com as diferentes comunidades através do mundo digital. Esta partilha de conhecimento pode ser muitas vezes facilitada através de sistemas de recomendação.

Um sistema de recomendação não é mais que uma análise das preferências de um utilizador com o propósito de recomendar pontos de interesse que vão de encontro às suas necessidades de informação. No contexto desta tese, foi implementado um sistema de recomendação cuja principal função é a partilha do espólio da UP, destacando pontos de interesse. Nesta tese, o principal obstáculo que se enfrentou foi o próprio estado embrionário do projeto Museu Digital da Universidade do Porto, designadamente a falta de dados relativos aos possíveis utilizadores que queiram explorar o espólio. Assim, esta tese apresenta um sistema de recomendação baseada na sugestão de pontos de interesse em sequência, através de um algoritmo de mineração de sequências frequentes, o SPADE. Este é um trabalho exploratório que tem em conta a possível expansão do projeto com a anexação de novos pontos de interesse e um histórico das visitas efetuadas pelos utilizadores e até mesmo o seu perfil.

Palavras-chave: sistema de recomendação, mineração sequências frequentes, ponto de interesse (POI), SPADE

Abstract

The exchange of historical, cultural and scientific heritage is nowadays accessible to global peoples thanks to information and communication technologies. Porto's University project *Museu Digital da Universidade do Porto* aims to share its assets with the different community's trough digital means. This method to share knowledge can be sometimes eased through recommender systems.

The main purpose of a recommender system is to suggest items to a user according to its needs and preferences. This thesis implements a recommender system in which the main purpose is to share the Porto's University assets, highlighting points of interest. The early stage of the project *Museu Digital da Universidade do Porto* was the main obstacle of this thesis, due to the lack of user's data. Therefore, this thesis presents a recommender system based on sequential data mining algorithm SPADE which suggests points of interest as a sequence. This is an exploratory work able to be adapted to the possible development of the project, with new points of interest data, the record of historical visit data and user profile data.

Keywords: recommender system, sequence pattern mining, point of interest (POI), spade

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e símbolos

Lista de abreviaturas

App	Aplicação para Redes Móveis
CSV	*Comma-Separated Values*
GSP	*Generalized Sequential Pattern Mining*
kNN	*k-nearest neighbors*
LIT-*PrefixSpan*	*Location-Item-Time PrefixSpan*
MDUP	Museu Digital da Universidade do Porto
NA's	Dados em falta
POI(s)	Ponto(s) de interesse
RGPD	Regulamento Geral de Proteção de Dados
SR	Sistema de Recomendação
UP	Universidade do Porto

Lista de símbolos

BD	Base de dados de rotas $< IDn\{In, Tn\} >$
I	Conjunto de itens/POIs que podem ser recomendados a um utilizador **u**
i	Item/POI sugerido pelo sistema de recomendação ao utilizador **u**.
$\boldsymbol{\theta}$	Conjunto de subsequências $< sequenceIDn\{Itemn\}[Tc'] >$ a serem recomendados
$\boldsymbol{qU}$	Pedido do utilizador sobre os Itens qi que pretende visitar
$\boldsymbol{qT}$	Pedido do utilizador sobre o Tempo qt que pretende dispensar
$\boldsymbol{\sigma}$	Um limiar de suporte mínimo
$\boldsymbol{s1}$	Conjunto de subsequências com suporte maior que σ
$\boldsymbol{\tau}$	Todos In com suporte maior que δ
$\boldsymbol{\tau'}$	Base de dados de transações $< sequenceIDn\{eventIDn, itemn\} >$
U	Conjunto de utilizadores que avaliaram previamente **I**
u	Utilizador que requer uma recomendação de um item/POI do conjunto **I**.

Índice de conteúdo

1. Introdução

1.1 Enquadramento do projeto e motivação

A Universidade do Porto dispõe hoje de um conjunto de espólios nos seus museus que podem e devem ser colocados à disposição e usufruto dos mais variados públicos, não se confinando, pois, o seu interesse à comunidade académica. O potencial histórico, cultural e científico desses acervos é um valor em si mesmo que merece um novo olhar sobre como utilizar esses materiais, difundindo-os e tornando-os acessíveis quer às diversas comunidades científicas, quer ao público em geral aproveitando as novas tecnologias de informação e comunicação para ampliar a sua utilização.

Na origem do projeto do Museu Digital da Universidade do Porto (MDUP) evidenciou-se o propósito de salvaguardar as coleções, pessoas e percursos entre as várias unidades museológicas e o seu contexto, mas também o objetivo de aproveitar esse património para fomentar a troca do conhecimento dentro e fora das paredes da Universidade do Porto. Ou seja, com o MDUP facilita-se, de forma sistematizada e relacionada, o acesso e a troca de conhecimento numa plataforma virtual em que se apresentam pontos de interesse (POIs) ou percursos "culturais" que possam ir ao encontro da procura de diferentes tipos de utilizadores, sejam estudantes, famílias, professores, etc.. Mas também turistas que deambulam pela cidade e que poderão beneficiar assim de um roteiro adequado ao interesse próprio sobre determinada área que eventualmente tenham (Pinto et al., 2016).

O projeto do MDUP arrancou com a criação de duas aplicações. Uma que conta com o apoio da comunidade estudantil para identificar os POIs e roteiros digitais referentes ao património universitário - #IWASHERE (parceria com a Faculdade de Engenharia da Universidade do Porto, e a unidade curricular de LGP118) e outro que inclui conteúdos acerca de locais que se relacionam com a Engenharia na região norte de Portugal, como, por exemplo, edifícios, pontes, minas, parques urbanos, ruas com nomes de engenheiros emblemáticos, etc. - #GPSEngenharia (parceria do MDUP com a Ordem dos Engenheiros Região Norte).

Os POIs e roteiros criados nestas aplicações constituíram, assim, um ponto de partida para esta tese de mestrado, ainda que não tenham sido objeto para a elaboração desta tese. Dado que o projeto se encontra numa fase embrionária ainda não há perfis de utilizadores disponíveis para se construir um sistema de recomendação (SR) completamente eficaz. Esta escassez de dados vai condicionar a escolha dos métodos de sistemas de recomendação, tal como a escolha do algoritmo mais adequado para esse fim. Uma forma de ultrapassar

estas condicionantes forçará a que se tome a opção de ter um maior enfoque no item e menos na similaridade entre utilizadores, dada a falta de dados sobre a avaliação dos itens por parte dos utilizadores.

Assim, nesta tese pretende-se investigar como definir um percurso de navegação que consiga prever e fornecer recomendações adequadas a um utilizador cujo perfil ainda não foi definido – problema *cold-start*, e que seja de certa forma adaptável à integração de novos dados sobre utilizadores, POIs e percursos na aplicação para redes móveis (app). Pretende-se, assim, contribuir para um projeto em crescimento que visa uma maior e melhor integração do MDUP na vida da cidade, numa aproximação virtuosa entre a Universidade e a sua cidade e o país.

1.2 Problemas, objetivos e resultados esperados

O objetivo desta dissertação é desenvolver um sistema de recomendação para um conjunto de rotas recolhidas pelo projeto do MDUP que possa ser facilmente adaptado a novos percursos.

Assim, para atingir esse objetivo definiram-se os seguintes objetivos específicos:

- Definir os requisitos essenciais para um sistema de recomendação para percursos culturais que estejam enquadrados nos objetivos do projeto da MDUP;
- Com base na revisão de literatura, selecionar algoritmos adaptados a sistemas de recomendação para percursos culturais e, ao mesmo tempo, à pouca disponibilidade de dados;
- Testar o algoritmo para os dados e rotas disponíveis, através de um software de programação adaptado à análise de dados e sistemas de recomendação, passíveis de serem adaptados ao crescimento do projeto.

A figura 1 apresenta o esquema da árvore de objetivos para esta dissertação.

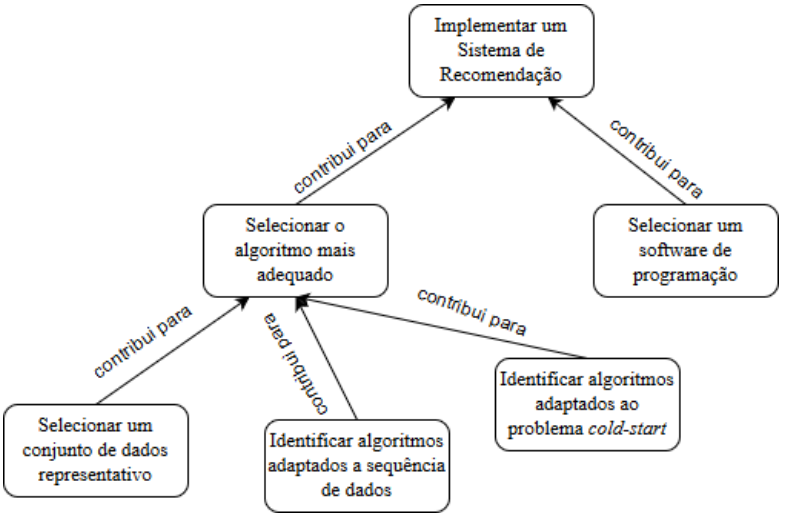


Figura 1 - Árvore de Objetivos

1.3 Abordagem metodológica

Nesta tese, optou-se por seguir uma metodologia exploratória dada o estado embrionário do próprio projeto e por não se tratar de uma metodologia fechada, permitindo antes a sua adequação a outras finalidades ao longo do tempo.

O método de investigação está dividido em três secções diferentes: revisão da literatura, seleção dos dados a serem utilizados, seleção do algoritmo para fazer a recomendação:

- Para fundamentar e realizar este trabalho procedeu-se à pesquisa do estado da arte em 2 bases de dados (Scopus e Science Direct) nas quais se selecionaram artigos relevantes entre as datas 2000 e 2018. Todos os documentos não recuperados (foi impossível aceder ao texto completo) foram tratados como não relevantes. Além da pesquisa de artigos, foram consultados três livros de referência ligados aos sistemas de recomendação e à mineração de dados. A seleção de documentos entre 2000 e 2018 teve como objetivos recuperar estudos atuais com novas técnicas de mineração de dados para aplicação nos diversos sistemas de recomendação e encontrar um ponto evolutivo e adaptativo das diferentes técnicas à evolução das tecnologias, como os *smart phones*.
- Os dados recolhidos pelos projetos #IWASHERE e #GPSEngenharia desenvolvidas no projeto do MDUP contêm informação sobre os POIs a nível de contexto, nome, morada, descrição adicional, tipo de POI (pessoas; objeto; local indoor/outdoor;

evento; narrativa; latitude/longitude, sendo que o percurso cultural, a rota, terá que incluir uma sequência de POIs). Dada a escassez de dados, optou-se por construir um sistema de recomendação baseado na sugestão de POIs mais frequentes numa determinada rota, procurando-se desta forma colmatar o facto de não existirem dados de utilizadores ou *rankings* dos POIs. Também devido à indisponibilidade dos dados durante a execução da tese optou-se por escolher como método o artigo de Tsai et al. (2015) devido à similaridade com o os dados do projeto do MDUP.

- Para a elaboração do conjunto de testes do sistema de recomendação serão utilizados algoritmos capazes de lidar tanto com a sequência de POIs, como com a falta de informação sobre o perfil do utilizador. Com base na revisão de literatura e o objetivo pretendido selecionou-se o algoritmo SPADE para determinar subsequências frequentes de POIs nas rotas disponíveis. A seleção deste algoritmo também esteve relacionada com a sua disponibilidade no software de programação R utilizado nesta tese.

1.4 Estrutura da dissertação

A tese está dividida em cinco capítulos:

No primeiro capítulo apresenta-se a contextualização, definição dos objetivos propostos e a metodologia para atingir esses mesmos objetivos.

No segundo capítulo é apresentada uma revisão de literatura (*State of the Art*) em que se indicam os artigos considerados de maior referência sobre os conceitos importantes nos sistemas de recomendação. Nestes artigos apontam-se documentos relevantes nos percursos culturais e problemas de *cold-start*, descrevem-se os diferentes sistemas de recomendação e problemas associados. Em seguida, apresentam-se algumas métricas usadas para avaliar a eficiência do sistema de recomendação.

No capítulo três descrevem-se sucintamente os dados disponíveis pelo projeto MDUP, o método proposto para obter uma sequência de recomendação baseada nesses dados, assim como a indicação do software de programação utilizado neste trabalho.

No quarto capítulo descreve-se a metodologia, no qual é apresentada uma análise exploratória dos dados e os resultados obtidos.

No último capítulo serão sumarizadas as maiores contribuições desta tese, tal como recomendações para trabalhos futuros.

As últimas páginas são dedicadas a referências bibliográficas e anexo.

2. Revisão de Literatura

O Conselho Internacional de Museus define o conceito de museu como

> "uma instituição permanente sem fins lucrativos, ao serviço da sociedade e do seu desenvolvimento, aberta ao público, que adquire, conserva, investiga, comunica e expõe o património material e imaterial da humanidade e do seu meio envolvente com fins de educação, estudo e deleite." (ICOM)

Diferenciadores na cultura dos povos e singularmente valiosos, os museus constituem-se como valores em si mesmo, mas é na apropriação por parte dos públicos, quer a nível de transmissão de conhecimento ou apenas reserva e guarda de coleções, que cumprem o seu maior papel, sendo, por isso fundamental promover a acessibilidade aos seus espólios e a sua divulgação.

Os sistemas de recomendação são soluções que se adequam a contextos culturais como os museus, muito embora sejam também globalmente seguidos a nível do comércio. São soluções que podem trazer grandes benefícios a nível de transmissão de conhecimento, não só sobre o objeto em si, seja ele qual for, como para usufruto de agentes culturais e partilha com os diferentes públicos para os quais trabalham (Solima et al., 2016). Concretamente, permitem que o visitante, quer cidadãos locais ou turistas, tenha uma experiência mais vivida e enriquecida do local que visita.

Atualmente, as várias plataformas ligadas ao turismo têm sistemas de recomendação que apoiam e dão sugestões sobre o local a visitar, ou até mesmo um percurso, sugestões estas adaptadas ao tipo de utilizador, como por exemplo crianças, famílias ou grupos escolares. Estes sistemas de recomendação têm um importante papel na tomada de decisão do utilizador na sua escolha de um itinerário, na decisão do tempo que vai gastar em cada local que pretende visitar, ou na sequência de lugares a visitar (Borràs, et al., 2014; Cai, et al., 2018).

2.1 Sistemas de Recomendação

Um sistema de recomendação é uma ferramenta que apoia a tomada de decisão de um utilizador ao fazer sugestões (Bobadilla, et al., 2013), orientando-o nos mais diversos campos: comprar livros, escolha de filmes que o podem interessar particularmente, sítios que desejaria e poderá visitar (Bobadilla, et al., 2013; Borràs, et al., 2014).

Os sistemas de recomendação usam vários tipos de informação para prever o que é que um utilizador quererá no futuro. A eficiência da recomendação resulta de um equilíbrio entre

vários fatores:

1. Relevância: o utilizador quer receber sugestões que acha interessantes e que são importantes na sua perspetiva, o que aumentará a probabilidade de vir a seguir a recomendação e comprar o produto.
2. Novidade/inovação: o utilizador espera recomendações novas de produto, e não algo que já tenha visto no passado.
3. Acaso: o utilizador, com uma dose de sorte, consegue descobrir por acaso algo que nunca tinha visto. É mais do que descobrir algo novo dentro do género que gosta, é encontrar algo latente no seu gosto, interesse e preferência, sugerindo-lhe algo que é inesperado. Trata-se de um fator que pode gerar benefícios ao nível da diversidade das vendas e lançar até novas tendências, indo ao encontro de interesses não revelados desse utilizador. Se é certo que nesse acaso haverá, por certo, sugestões irrelevantes para o utilizador também é verdade que no leque das opções desse acaso haverá sugestões que o vão influenciar positivamente.
4. Diversidade de recomendações: o utilizador espera alguma diversidade nas recomendações porque é bem provável que se receber inúmeras recomendações sobre um único tópico se possa sentir aborrecido, embora seja o tópico pelo qual manifesta preferência.

Existem, por outro lado, outros objetivos complementares aos objetivos definidos em cima tendo em vista a fidelização do utilizador. Por exemplo, um sistema de recomendação que faz sugestões periódicas induz a que o utilizador volte, em princípio, a utilizar o serviço. A experiência do utilizador pode, também, ser melhorada sempre que perceba o motivo por que está a receber novas recomendações, ao cruzá-las com o histórico das suas escolhas (Bobadilla, et al., 2013; Aggarwal, 2016).

Para ser possível sugerir itens, percursos, etc. o sistema de recomendação tem de ser capaz de prever as necessidades do utilizador no futuro. E para isso recorre-se muitas vezes à mineração de dados e ao uso de algoritmos. O princípio básico de um algoritmo de recomendação é o de que existe uma correlação significativa entre o utilizador e o produto/serviço que seleciona. Por exemplo, um utilizador que vê um filme de ação pode estar mais interessado num thriller do que num documentário sobre natureza (Aggarwal, 2016).

2.2 Tipos de Sistemas de Recomendação

Os modelos típicos dos sistemas de recomendação trabalham com dois tipos de dados. Associados a técnicas de filtragem colaborativa estão os dados sobre iteração utilizador-item, como, por exemplo, avaliações sobre um produto ou comportamento "de compra" do utilizador. Por outro lado, ligados a técnicas de recomendação baseadas em conteúdo estão os atributos dos utilizadores e itens, como, por exemplo, o perfil de utilizador ou palavras-chave associada ao produto (Aggarwal, 2016).

Os sistemas de filtragem colaborativa são os mais desenvolvidos e implementados nos sistemas de recomendação, o que se pode ficar a dever ao facto de terem surgido primeiro (Bellogín et al., 2017). Além destes dois tipos de sistemas de recomendação, há autores que também identificam outros: sistemas de recomendação baseados em conhecimento, demográficos e sistemas de recomendação híbridos que incluem, por exemplo, dados temporais (Burke, 2002; Bobadilla, et al., 2013; Bellogín et al., 2017).

Os subcapítulos seguintes falam um pouco dos sistemas de recomendação mais frequentes.

2.2.1 Filtragem Colaborativa

O método de filtragem colaborativa usa o *ranking* de um utilizador, o seu histórico de compras, ou avalia a similaridade de gostos entre utilizadores para prever o que é que poderão vir a ser as suas escolhas futuras. Os utilizadores ao avaliarem os produtos/serviços estabelecem um padrão e a sugestão é dada de acordo com o item mais popular, ou porque o utilizador fez escolhas semelhantes no passado. (Aggarwal, 2016; Bellogín et al., 2017). Este método evidencia várias vantagens, como, por exemplo, sugerir itens surpreendentes. Todavia, apresenta desafios que se prendem com a dispersão de dados (*sparsity*) e com problemas de escala (Park, et al., 2012; Borràs, et al., 2014).

As técnicas deste método podem ser classificadas em duas categorias: *model-based* e *memory-based* (ou heurístico). A primeira constrói padrões das diferentes iterações utilizador/item para gerar previsões automáticas. A segunda baseia-se na similaridade entre utilizadores nas suas escolhas dos itens, ou na similaridade dos próprios itens.

O algoritmo de referência aplicado na técnica *memory-based* é *o k-Nearest Neighbors* uma vez que explora a similaridade para classificar os utilizadores/itens, através do *ranking*, e utiliza os resultados mais próximos, os resultados vizinhos, para fazer a recomendação (Park, et al., 2012; Bellogín et al., 2017). A grande vantagem do algoritmo reside na simplicidade do uso e na precisão dos resultados, sendo possível fazer boas previsões e recomendações ao utilizador (Bobadilla, et al., 2013).

Tran et al. (2016) aplica um algoritmo baseado em modelos probabilísticos: *Markov Random Fields*, que fundamentalmente se baseia num grafo que une variáveis aleatórias. Os autores propõem um método em que os nós representam as preferências de um utilizador ou item, e as ligações representam a dependência entre itens e entre utilizadores. Esta técnica é bastante robusta e precisa, permitindo estimar automaticamente a relação item-item e utilizador-utilizador, mas contém várias desvantagens a nível de complexidade e dispersão de dados.

Na realidade, a filtragem colaborativa é aplicada em várias áreas. A nível de marketing online temos, por exemplo, a Amazon, que faz reger recomendações com base na similaridade dos itens e no que o utilizador comprou no passado, ou seja o seu histórico, fazendo uma comparação do perfil de consumo desse utilizador com a popularidade de itens semelhantes (Aggarwal, 2016).

Na área do turismo, o método é também é bastante aplicado. É disso exemplo a utilização do algoritmo *k-means* para obter uma série de sequências de pontos de interesse turísticos, para agrupar características demográficas dos utilizadores e ainda para agrupar os utilizadores de acordo com a sua avaliação sobre as recomendações feitas (Borràs, et al., 2014).

2.2.2 Sistemas de Recomendação baseados em conteúdo

Os sistemas de recomendação baseados no conteúdo sustentam-se na análise dos atributos de um item. A combinação da avaliação de um item por um utilizador e o seu comportamento de compras são os dois fatores que se conjugam para chegar à recomendação de itens semelhantes (Aggarwal, 2016).

O método baseado em conteúdo calcula o grau de similaridade entre utilizadores e itens a serem recomendados. Quanto mais alto for o valor obtido mais robusta é a recomendação (Borràs, et al., 2014). No entanto, uma vez que compara o grau de similaridade, o método pode incluir itens que o utilizador já conhece, e pode não funcionar de forma eficaz devido à falta de conhecimento sobre os utilizadores – efeito *cold-start* (Park, et al., 2012; Borràs, et al., 2014).

Semeraro et al. (2012) propõem a adição de palavras-chave aos itens para que o utilizador as use e, assim, permitir que através do sistema de recomendação se possa inferir o seu perfil. Para análise da frequência das palavras-chave os autores aplicaram o algoritmo *Näive Bayes* que se mostrou eficiente na classificação dos itens/utilizadores sendo mais simples de aplicar do que, por exemplo, o algoritmo *Support Vector Machines*. A

recomendação é feita, depois, de acordo com a similaridade de outras palavras-chave e o histórico de avaliação.

Binucci et al. (2017) criou um algoritmo baseado em grafos para criar um sistema de recomendação para atrações turísticas. A arquitetura do sistema começa com a recolha de informação de várias fontes e cria uma descrição plausível para os POIs, que, por sua vez, é inserida num modelo de conhecimento que constrói o perfil do utilizador. O resultado é uma lista de POIs similares aos do perfil do utilizador.

É, portanto, um sistema de recomendação que tem a vantagem de se adaptar bem à mudança de preferências e o *feedback* implícito é suficiente para fazer boas recomendações. No entanto pode sofrer problemas de *cold-start* a nível do utilizador ou itens, como se referiu acima. De facto, a qualidade das recomendações está dependente um grande volume de dados históricos e as recomendações são, de uma maneira geral, estáticas, ou seja, as recomendações são pouco diversificadas (Burke, 2002).

2.2.3 Sistemas de Recomendação baseados em conhecimento

Os sistemas de recomendação baseadas no conhecimento são particularmente úteis quando os itens não são comprados com muita frequência, veja-se os casos de um apartamento ou de um carro. É um sistema que utiliza o conhecimento do domínio para relacionar os requisitos do utilizador com as propriedades do item.

Este tipo de prolema é associado também ao problema *cold-start*, precisamente por não haver avaliações suficientes para o processo de recomendação. Além de que o comportamento e preferências do consumidor pode ir-se alterando com o tempo, sendo, pois, difícil capturar os interesses do utilizador apenas observando o histórico. Assim, é um sistema de recomendação que mais do que colocar em evidência o item, procura adequar os atributos com os requisitos específicos do utilizador (Aggarwal, 2016).

No *e-tourism* os sistemas de recomendação baseados em conhecimento são normalmente apoiados em ontologias. Ruotsalo et al (2013) e Borràs et al. (2014) apresentam medidas de comparação da similaridade de duas atividades ou de dois utilizadores, baseadas em ontologias, nas quais fazem uma categorização dos itens.

2.2.4 Sistemas de Recomendação Demográficos

Os sistemas de recomendação demográficos fazem sugestões com base no perfil demográfico do utilizador, como por exemplo a idade, estudos, nacionalidade. As sugestões

destes sistemas têm como fundamento que os utilizadores que partilham o mesmo perfil demográfico estarão interessados nos mesmos itens. É normalmente um sistema que combinado com outros sistemas de recomendação, como por exemplo sistemas baseados em conhecimento, ou outro tipo de características ligadas ao contexto da recomendação, tornam os resultados das recomendações muito robustos (Aggarwal, 2016). Estes sistemas de recomendação, em combinação com outros métodos, minimizam o efeito *cold-start* já que, à partida, estabelecem o perfil do utilizador (Borràs, et al., 2014; Binucci et al., 2017).

A tabela 1 apresenta um sumário das características para os quatro sistemas de recomendação referidos acima, sistematizados por Burke (2002).

Na tabela, **I** é o conjunto de itens que podem ser recomendados ao utilizador **u**, através da avaliação que outros utilizadores **U** fizeram. E **i** é o item sugerido pelo sistema de recomendação ao utilizador **u**.

Tabela 1 - Sumário das características dos sistemas de recomendação

Sistema de Recomendação	Base	Input	Processo
Filtragem Colaborativa	Avaliações de **U** para itens em **I.**	Avaliações de **u** para itens em **I**	Identificar utilizadores em **U** que são semelhantes a **u**, e extrapolar pelas suas avaliações o item **i**.
Baseada em conteúdo	Características dos itens em **I.**	Avaliações de **u** para itens em **I**	Classificar o comportamento de avaliação de **u** e utilizar em **i.**
Demográfica	Informações demográficas sobre **U** e as suas avaliações a **I.**	Informações demográficas de **u**	Identificar utilizadores demograficamente semelhantes a **u**, e extrapolar pelas suas avaliações o item **i**.
Baseada em conhecimento	Características dos itens em **I**. Conhecimento de como esses itens se adequam às necessidades do utilizador.	Descrição das necessidades ou interesses de **u**.	Inferir a correspondência entre **i** e as necessidades de **u**.

in Burke. "Hybrid Recommender Systems: Survey and experiments". User Modeling and User-Adapted Interaction, 12(4) (2002):2

2.2.5 Sistemas de Recomendação Híbridos

Os sistemas de recomendação híbridos combinam várias técnicas de recomendação ou combinam o resultado de diferentes sistemas de recomendação para produzir uma única

sugestão. Este tipo de método tenta tirar partido das vantagens de cada um deles e evitar as desvantagens (Burke, 2002).

Burke (2002) considera que existem as seguintes técnicas de hibridização:

- Ponderada: os resultados de várias técnicas de recomendação são combinados para produzir uma única recomendação
- Alternada: o sistema vai revezando métodos de sistemas de recomendação, dependendo da situação
- Mistura: sugestões de diferentes sistemas de recomendação são apresentados ao mesmo tempo.
- Combinação de características: características de diferentes fontes de diferentes sistemas de recomendação são incluídas num único algoritmo de recomendação.
- Cascata: um sistema de recomendação afina a recomendação resultante de outro.
- Aumento das características: o resultado de uma técnica é usado como input de outra.
- Meta-level: o modelo apreendido por um sistema de recomendação é usado como input de outro.

Benouaret et al. (2015), apresentam um sistema de recomendação para dispositivos móveis que se adapta ao perfil do utilizador e ao contexto, local, tempo, etc.. Os autores propõem um sistema de recomendação híbrido que aproveita o potencial dos sistemas de recomendação de filtragem colaborativa com aqueles que são os baseados em conteúdo. Na abordagem baseada em conteúdo, os autores exploram o vocabulário associado aos itens para determinarem a similaridade entre eles. E, numa perspetiva de filtragem colaborativa, usam a correlação de *Pearson* para analisarem a semelhança dos itens avaliados, e o algoritmo *k-nearest neighbors* (kNN) para determinarem a similaridade dos perfis de utilizadores com base nos dados demográficos. Estes dois métodos são unificados num único resultado - uma visita guiada adequada ao utilizador.

No projeto Turist@, Batet et al., (2012) providenciam recomendações personalizadas de atividades. Utilizam informação armazenada no perfil do utilizador de maneira a descobrirem que tipo de eventos podem ser selecionados, evitando uma oferta supérflua de atividades. Os autores apostaram um sistema de recomendação híbrido que combina um sistema baseado em conteúdo com filtragem colaborativa. O objetivo do sistema baseado em conteúdo era avaliar a relação entre um utilizador e as descrições das diferentes atividades. Para isso, cada item e perfil de utilizador foi representado por um vetor e aplicaram-se várias métricas de distância para determinar a similaridade entre dois vetores.

No método de filtragem colaborativa, aplicaram um algoritmo *cluster* (ClusDM), que é apenas repetido a cada entrada de 10 novos utilizadores, devido ao custo da sua implementação.

Ver-se-á, de seguida, algumas técnicas adicionais aos sistemas de recomendação híbridos que têm como objetivo conferir-lhes mais contexto.

Sistemas de Recomendação sensíveis à dimensão tempo

O tempo é um fator que pode determinar a eficiência de um sistema de recomendação. Por um lado, os interesses de um utilizador podem alterar-se ao longo do tempo e, por outro, a recomendação pode estar dependente da altura do ano, ou do dia - a recomendação de umas luvas de lã em pleno Verão seria um exemplo.

A variação dos interesses do utilizador ao longo do tempo pode ser controlada incorporando o fator tempo nos sistemas de recomendação por filtragem colaborativa. A recomendação de um item ao longo do dia ou ano pode ser encarada como um caso especial de sistemas de recomendação baseadas em conteúdo (Aggarwal, 2016).

Campos et al. (2014) fazem uma revisão de literatura na qual exploram os vários contextos aplicados a esta dimensão. Por exemplo, recomendações sazonais vs. recomendações durante um dia de visita. Dissertam também sobre soluções encontradas na literatura para parametrizar o tempo nos diversos sistemas de recomendação.

Sistemas de Recomendação baseados na localização

Os sistemas de recomendação baseados na localização do utilizador são cada vez mais frequentes, como sugerir um restaurante com base na localização e na avaliação do utilizador de outros restaurantes. Há dois tipos de localização que são considerados em sistemas de recomendação:

1. Localização específica do utilizador (*locality*): a localização física do utilizador tem normalmente um impacto na recomendação. A recomendação de tripas à moda do Porto em Lisboa seria quase absurda;
2. Localização específica do item. A localização geográfica do item, por exemplo um restaurante, pode ter impacto na relevância da sugestão (Aggarwal, 2016).

Cai et al., (2018) apresentam um sistema de recomendação baseado em palavras-chave associadas a fotografias e apresentam uma recomendação através de um algoritmo que se

baseia na semântica dessas palavras. A solução apresentada pelos autores tenta lidar com o tipo de local, restaurante, parque, ou a meteorologia, entre outros. Ou seja, em vez de recomendar um local, recomenda categorias de locais, através do histórico do movimento das pessoas, através de uma mineração constante das fotografias georreferenciadas.

Sistemas de Recomendação baseados na comunidade

Este tipo de técnica está associada à ideia de que o utilizador tende a gostar de algo se for sugerido por um amigo, mais do que se for sugerido por alguém que não conhece. São técnicas aplicadas sobretudo nas redes sociais em que se avaliam as relações sociais entre as pessoas e as preferências da rede de amigos, através de análise de gostos ou seguidores, por exemplo (Ricci, et al., 2011).

2.2.6. Sistemas de recomendação em sequência

Os sistemas de recomendação em sequência são nem mais nem menos do que a recomendação lógica de uma série de itens. Por exemplo, no caso dos percursos culturais podem ser encarados como a recomendação de uma sequência de POIs, de um museu, passando para um concerto e a seguir uma pausa num restaurante. Este tipo de SR pode ser particularmente útil pois integra várias características, como geografia, cultura, lazer, etc..

Por exemplo Kurataa et al. (2014) propõem uma app, CT-Planner, que oferece planos turísticos que são refinados a partir das preferências do utilizador, duração, relutância para andar, etc.. Para sugerir uma rota, a app faz apenas duas perguntas no início do percurso, o destino e o modo preferido de viajar, e as condições do tour, duração, hora inicial, dia da semana, velocidade a andar e a relutância em andar.

No entanto, não há muita literatura sobre recomendações para sequência de utilizadores ou itens. Muitas organizações optam por recolher informação sobre sequência de utilização através de *logs* de atividade (Bellogín et al., 2017) ou usando uma técnica do mineração de dados em sequência com o algoritmo *Apriori* e o *Generalized Sequential Pattern Mining* (GSP), que determina a frequência de um conjunto de itens. Fomenta, em particular, a seleção dos itens não frequentes e cria um super conjunto de itens, evitando contagens desnecessárias de itens que se sabe não serem frequentes (Ricci, et al., 2011; Aggarwal, 2015).

Recentemente alguns autores exploraram dois métodos, por vezes combinados, *Markov Chains* e *Matrix Factorisation Algorithms* para analisarem o padrão de escolha dos

utilizadores. São técnicas competitivas devido à capacidade de funcionarem com escassez de dados.

Os autores He et al. (2016) defendem que a combinação dos dois algoritmos são capazes de lidar com as iterações utilizador-item e item-item e prever a sequência de comportamentos a partir de dados colaborativos. Fundamentam que, ao combinar com um algoritmo *Factored Item Similarity Models,* permite que a recomendação ao utilizador seja baseada exclusivamente em quão semelhantes são os itens que foram já consumidos/gostados por esse utilizador. A combinação com *Markov Chains* permite capturar o padrão da sequência.

Tsai et al. (2015) apresenta um estudo de um parque temático onde o utilizador tem a recomendação de locais, itens e o tempo como sequência. Aplicaram o algoritmo combinado com sequência *Location-Item-Time PrefixSpan* (LIT-PrefixSpan) para sugerirem um percurso de acordo com os requisitos e limitações do visitante (em termos de gostos, tempo disponível). O ponto de partida é a definição das preferências do utilizador e o intervalo de tempo disponível, obtendo-se como recomendação a definição de um percurso, de acordo com os requisitos do utilizador.

Bellogín et al. (2017) propõem uma técnica que compara a similaridade das sequências comportamentais de utilizadores ou itens. Os autores adaptaram um algoritmo usado para determinar a sequência de caracteres – algoritmo *Longest Common Subsequence* – tendo as seguintes transformações: representar o utilizador como sequência, definir um valor limite para apresentar sequências pouco semelhantes e duas normalizações para serem aplicadas ao resultado do algoritmo. Os autores declararam que este algoritmo pode ser aplicado no comércio eletrónico, com bons resultados na precisão e recuperação de informação (Bellogín et al., 2017).

Além destas opções mencionadas por diferentes autores, o algoritmo probabilístico *Hidden Markov Models* também pode ser aplicado para analisar o padrão de uma sequência e prever o próximo passo. A vantagem no uso deste algoritmo em comparação com os *Markov Models* é o facto de trabalhar com variáveis, na sequência de dados, ocultos ao utilizador (Aggarwal, 2015).

Além do algoritmo *Prefix-span*, útil para determinar sequências mais frequentes, o algoritmo SPADE usa um formato vertical dos dados onde se associa para cada sequência a lista de itens. O algoritmo começa por identificar as subsequências com um item e a partir desses resultados continua a tentar encontrar sequências de dois itens, e por aí fora, até não ser possível encontrar subsequências frequentes para um dado suporte (Zaki, 2001).

2. 3 Problemas associados aos Sistemas de Recomendação

2.3.1 *Cold-start*

Associado ao conceito de sistema de recomendação está o problema do novo utilizador ou novo item – *cold-start*. Neste caso, o sistema de recomendação não sabe o que sugerir dado que o utilizador ainda não definiu as suas preferências, ou o item ainda não foi avaliado por ninguém. Os métodos baseados em conteúdo são mais robustos do que os baseados em filtragem colaborativa, uma vez que os primeiros apenas têm um problema de *cold-start* quando há um novo utilizador (Aggarwal, 2016).

Há uma série de técnicas na literatura que tentam solucionar este problema inicial de muitos sistemas de recomendação. Bobadilla et al. (2013) apresentam soluções que reportam a técnicas de clustering, tanto para utilizadores como para itens, para melhorar a previsão, ou palavras-chave. Lika et al. (2014), contruiu um modelo para ser adaptado a abordagens de filtragem colaborativa constituída por três passos: recolha de dados demográficos, considerando que pessoas com o mesmo background têm gostos semelhantes; cálculo da semelhança entre cada utilizador incluído nessa categoria demográfica; previsão através da qual se comparam as combinações encontradas na fase anterior com as avaliações de outros utilizadores para os itens a recomendar. Hernando et al., (2017) apresentam uma solução baseada num modelo probabilístico. No fundo, deixam que os novos utilizadores infiram por si mesmos a sua própria recomendação. Através de uma árvore de decisão conseguem prever qual é a probabilidade de gostar mais do item A ou B. Por outro lado, Peng, et al. (2016) sugerem executar o algoritmo n-dimensional *Markov Random Field* antes do algoritmo *Matrix Factorization*, para considerar atributos como idade, ocupação, etc., para um novo utilizador do sistema de recomendação. Os autores obtiveram melhores predições quando as compararam com predições de modelos de regressão linear (*Pairwise Preference Regression* por exemplo), ou algoritmo *Random Forest*.

2.3.2 Dispersão de dados

Este problema ocorre quando o número de avaliações dos utilizadores é pequeno em comparação com o número total de itens. Por isso, a probabilidade de encontrar um utilizador que avalie o mesmo item é muito baixa para fazer uma boa estimativa (Borràs, et al., 2014). Normalmente nestes casos apenas os itens mais populares são avaliados e a recomendação de itens que não são avaliados pode não ser possível, mesmo que em teoria fossem boas sugestões (Benouaret, et al., 2015).

Este é um problema associado em regra a técnicas de filtragem colaborativa, uma vez que usam algoritmos de agregação, como kNN, para recomendar itens de acordo com a avaliações semelhantes. Para diminuir este impacto de dados dispersos Bobadilla et al. (2013) mencionam métodos como *Matrix Factorization* adequados para trabalhar com grande volume de dados e adequados, também, a problemas de escala. Outro método seria a combinação das técnicas *Latent Semantic Index* e *Singular Value Decomposition* mas, normalmente aplicados quando a preferência dos utilizadores não se altera com o tempo (Bobadilla, et al., 2013).

2.3.3 Escala

O problema de escala acontece quando os dados são muitos, como, por exemplo, a quantidade de filmes e utilizadores do Netflix. Aqui, a utilização de algoritmos como kNN no caso dos sistemas baseados em filtragem colaborativa torna-se muito lenta (Bobadilla, et al., 2013).

2.3.4 Resultados demasiado especializados

Resultados especializados são, no fundo, recomendações de itens, locais ou percursos demasiado semelhantes com aqueles que o utilizador gostou, selecionou no passado. Como referido anteriormente, normalmente é aconselhado haver alguma diversidade nas recomendações, principalmente na perspetiva de um turista (Benouaret, et al., 2015).

Os sistemas de recomendação baseados em conteúdo são muito sensíveis a este tópico, uma vez que o seu objetivo é precisamente detetar as semelhanças entre itens que partilham o mesmo atributo ou característica (Park, et al., 2012). Seria o caso de uma pessoa que ouça música clássica, que terá os atributos definidos nesse sentido, e será pouco similar a uma pessoa cuja preferência seja o metal. Por isso é importante ter algum tipo de novidade e acaso nas recomendações. Para ultrapassar isto pode sugerir-se, eventualmente, itens inesperados de acordo com as preferências do grupo em que que esse utilizador se insere (Aggarwal, 2016).

2.3.5 *Grey sheep*

Este é um problema associado aos sistemas de recomendação por filtragem colaborativa. A designação *Grey sheep* deve-se ao facto de se referir a um utilizador cujo perfil é díspar dos

restantes utilizadores já integrados, e torna-se difícil associar recomendações de utilizações similares (Borràs, et al., 2014).

2.3.6 Outros

Benouaret et al. (2015) acrescenta um problema muito particular aos museus que pode ser extrapolado noutras circunstâncias. Os autores sugerem um percurso cultural que cubra os interesses do utilizador no limite de tempo que ele tem para o fazer.

Batet et al., (2012) propõem um sistema de recomendação que impede uma recomendação em demasia, quando se propõe, designadamente, inúmeras atividades para um percurso turístico e mesmo demasiadas para um tempo limitado. O autor propõe um inquérito inicial ao utilizador através do qual se obtém a informação sobre o tempo que pretende disponibilizar para as atividades que procura e o tipo de atividade, língua, data do evento, etc..

A tabela seguinte sumariza os desafios dos quatro sistemas de recomendação identificados pela maioria dos autores.

Tabela 2 - Sumário dos desafios nos sistemas de recomendação

	Desafios dos Sistemas de Recomendação				
Sistema de Recomendação	*Cold-start*	Dispersão de dados	Escala	Resultados especializados	*Grey sheep*
Filtragem Colaborativa	✓	✓	✓		✓
Baseada em conteúdo	✓			✓	
Demográfico			✓	✓	
Baseada em conhecimento		✓		✓	

2.4 Avaliação do sistema de recomendação e da similaridade dos itens

A avaliação do sistema de recomendação é um passo essencial para determinar a precisão da decisão e a similaridade entre os utilizadores e/ou itens.

As técnicas de medição de qualidade usadas tradicionalmente são as avaliações das

previsões, das recomendações como bloco e das recomendações como uma lista hierárquica de sugestões. As métricas utilizadas tipicamente nos sistemas de recomendação são métricas que visam determinar quão exata é a previsão do que o utilizador procura: *mean absolute error*; *mean squared error*, *root of mean square error*, *normalized mean average* (Bobadilla, et al., 2013; Lika et al., 2014); ou métricas de classificação, *precision*, *recall* e F1 (Ricci, et al., 2011; Bobadilla, et al., 2013) ou métricas para avaliar a lista ordenada de recomendações, através do *Discounted Cumulative Gain* (Bobadilla, et al., 2013).

Entre as técnicas mais utilizadas para avaliar a similaridade entre itens encontra-se a correlação de *Pearson*, distância euclidiana (Batet, et al., 2012), *cosine*, *adjusted cosine*, *constrained correlation*, *mean squared differences*, e, mais recentemente, o coeficiente de *Jaccard* (Bobadilla, et al., 2013).

3. Metodologia

Nesta secção é apresentada uma breve descrição dos atributos dos dados a serem utilizados na fase de experimentação, os algoritmos utilizados para elaboração do sistema de recomendação, as métricas utilizadas para fazer a sua avaliação e uma breve descrição das tecnologias utilizadas.

3.1 Conjunto de Dados

A recolha de dados tem sido realizada pelo projeto MDUP contendo apenas as rotas e tempo associado a cada POI, com os seguintes atributos:

- IDRota Código identificador da rota.
- IDPOI Código identificador do POI
- Tempo Tempo gasto na visita em cada POI, para cada rota

Como foi referido, os dados consistem em rotas que podem conter um ou mais POIs. Por exemplo, uma rota pode conter a referenciação de várias igrejas que fiquem na mesma área. Ou pode existir uma rota cujo único POI é uma igreja (por exemplo a Igreja dos Clérigos).

Cada rota terá na sua sequência de POIs um tempo de visita associado. Este valor pode variar consoante as rotas, mesmo que se refira ao mesmo POI. Por exemplo, numa rota que inclua seis museus é possível que o tempo de visita em cada um deles seja menor do que uma rota que contenha apenas dois museus.

Dada a indisponibilidade destes dados por parte do projeto MDUP para esta tese, optou-se por adaptar os dados do artigo Tsai et al. (2015) ao cenário do projeto da MDUP.

3.2 Mineração de Sequências Frequentes

O objetivo deste trabalho é identificar e recomendar uma sequência de POIs a um utilizador utilizando uma série de técnicas de mineração de sequências. Esta sequência de POIs não é necessariamente uma rota existente na base de dados, mas sim a seleção de POIs encontrados em sequência.

Ou seja, para uma série de sequências $S\ \{s1, s2, \dots sn\}$, em que cada sequência contém uma transação de um conjunto de itens $I\ \{i1, i2, \dots in\}$ numa determinada ordem temporal, é possível identificar os padrões frequentes, associações e correlações entre os diferentes itens que cada sequência tem no seu "cesto".

Por exemplo, $(\{Museu\ A, Igreja, Coliseu\}, \{\ Museu\ A, Coliseu\})$ tem duas sequências com

três e dois itens cada uma, respetivamente. Os itens $\{Museu\ A\}$ e $\{Coliseu\}$ aparecem em ambas as sequências. A ordem temporal é um fator importante na determinação da transação, sob a forma de data ou apenas horas/minutos. A determinação das subsequências mais frequentes baseia-se na probabilidade de, no caso do exemplo, ao item $Museu\ A$ seguir-se a visita ao item $Coliseu$, ou seja, de $\{Museu\ A, Coliseu\}$ ser uma das subsequências possíveis das duas sequências mencionadas. As medidas utilizadas para avaliar essa probabilidade e devolver um certo número de subsequências são o suporte e confiança. O suporte de uma subsequência Y é definido como frequência em que Y aparece em S, um valor elevado significa que é uma subsequência cujos itens I aparecem com regularidade em cada sequência S. A confiança é traduzida como a percentagem de itens em $s1$ que podem também ser encontrados em $s2$. Assim, uma subsequência é denominada frequente quando ocorre mais vezes que o suporte mínimo definido. Uma subsequência frequente é máxima, suporte 100% se não é uma subsequência de qualquer outra sequência frequente. Por exemplo, $\{Museu\ A, Igreja\}$ é uma subsequência com suporte 50% pois é uma subsequência que depende de uma sequência maior, e com confiança 50% pois é uma subsequência que tem essa ordem de itens em 50% das sequências apresentadas.

Um requisito importante para usar esta técnica de mineração de dados em sequência é ter um algoritmo rápido e eficiente pois o tempo de processamento computacional dos dados tem um grande impacto quando se trabalha com um grande número de dados. Deste modo, selecionou-se o algoritmo SPADE, tanto pela disponibilidade na biblioteca do R, como pela sua maior eficiência face ao algoritmo GSP (Zaki, 2001). A alternativa era a seleção do algoritmo de mineração de sequências frequentes no software R o TraMineR (Trajectory Miner in R). Ao contrário deste algoritmo, o SPADE tem a particularidade de gerar subsequências respeitando as regras de associação, ou seja, considerando que os itens $\{Igreja, Coliseu\}$ aparecem sempre a seguir a $\{Museu\ A\}$ nas sequências de S, então as subsequências Y terão sempre a ordem $\{Museu\ A, Igreja, Coliseu\}$ e nunca $\{Igreja, Museu\ A, Coliseu\}$. Assim, assume-se nesta tese que o tempo cumulativo de visita das subsequências é representativo do tempo total numa visita que inclua os trajetos. Porque, se se considerar que a subsequência $\{Museu\ A, Igreja, Coliseu\}$ tem um percurso direto bem definido, não se pode dizer que ao percorrer a subsequência $\{Igreja, Museu\ A, Coliseu\}$ o percurso não seja maior.

3.2.1 Algoritmos para obter recomendações

Com uma abordagem mineração de sequências é possível identificar o padrão que ocorra vezes suficientes numa rota, dado pelo suporte, de maneira a que seja considerado de interesse para visitar. Indiretamente, esses padrões permitem recomendar os POIs mais populares e importantes. A figura seguinte apresenta a estrutura dos algoritmos aplicados para preparar o conjunto de dados para mineração de sequências.

Algoritmo 1. Algoritmo para preparação dos dados

Entrada:

BD Base de dados de rotas $< IDn\{In, Tn\} >$

δ Um limiar de suporte mínimo

Saída:

τ Todos In com suporte maior que δ

τ' Transação de $< sequenceIDn\{eventIDn, itemn\} >$

Método:

1: **para cada** $< IDn\{In, Tn\} > \in$ BD
2: **se** frequência $In > \delta$ **então**
3: adicionar $< IDn\{In, Tn\} >$ a τ
4: **senão** apagar $< IDn\{In, Tn\} >$
5: **fim se**
6: **fim**
7: **para cada** evento $Tn \in \tau$
8: calcular o tempo cumulativo Tc
9: adicionar Tc a τ
10: **fim**
11: **para cada** $< IDn\{In, Tcn\} > \in \tau$
12: transformar em transação $< sequenceID\{eventID, item\} >$
13: adicionar $< sequenceIDn\{eventIDn, itemn\} >$ a τ'
14: **fim**

Figura 2 - Pseudocódigo para a preparação dos dados para o algoritmo que gera subsequências

O input do algoritmo é a base de dados contendo um número de rotas IDn, com determinado número de POIs In e tempo de visita de associado Tn. O objetivo do Algoritmo 1 é, em primeiro lugar, remover os POIs que aparecem uma única vez (suporte mínimo δ menor que dois), por erro, ou porque são POIs tão específicos que não faz sentido que sejam sugeridos. Depois, calcula o tempo cumulativo Tc da visita de cada sequência de POIs, de maneira a assumir o tempo total de visita e percurso de uma sequência. O último passo é transformação dos dados em transação τ' de maneira a ser possível correr o algoritmo SPADE.

O Algoritmo 2 apresenta de uma maneira genérica o algoritmo para gerar subsequências de POIs dos dados preparados através do Algoritmo 1. Através de uma definição do suporte mínimo σ, o algoritmo percorre a transação τ' e procura encontrar subsequências de

tamanho 1. Se a frequência for maior que o suporte mínimo, o algoritmo tenta encontrar subsequências de dois itens, e por aí adiante.

Algoritmo 2. Algoritmo para gerar subsequências frequentes

Entrada:

τ' Base de dados de transações $< sequenceIDn\{eventIDn, itemn\} >$

σ Um limiar de suporte mínimo

Saída:

$s1$ Conjunto de subsequências ordenadas com suporte maior que σ

Método

1: **para cada** $sequenceIDn \in \tau'$
2: percorrer τ' para encontrar todos $\{itemn\}$ frequentes
3: **se** $suporte$(subsequência $\{itemn\}) > \sigma$ **então**
4: adicionar $suporte(\{itemn\}[eventIDn])$ a $s1$
5: adicionar o valor de σ a $s1$
6: **senão** eliminar a subsequência
7: **fim se**
8: **fim**
9: **para cada** subsequência $\in s1$
10: calcular o tempo cumulativo Tc' e adicionar a $s1$
11: calcular o comprimento da subsequência $freq$ e adicionar a $s1$
12: ordenar $s1$ de acordo com o suporte e o comprimento da subsequência
13: **fim**

Figura 3 - Pseudocódigo de um algoritmo de mineração de subsequências

Após o tratamento dos dados e da geração de subsequências segue-se a fase da recomendação dos POIs, de acordo com as preferências de visita do utilizador dentro do tempo disponível. O algoritmo 3 apresenta o conjunto de passos desde a geração de subsequências até à sugestão final. Parte-se do princípio que o utilizador sabe com antecedência qual POI qi pretende visitar e durante que espaço temporal qt. O objetivo é recomendar uma subsequência de POIs dentro da seleção POI e tempo de visita total de um utilizador. Se o algoritmo conseguir encontrar uma subsequência com o(s) POI(s) pretendido(s), então considera-se que a recomendação é eficaz. As sugestões θ são por fim ordenadas por suporte σ, número de POIs em cada subsequência $freq$ e, por fim, no caso de esses dois atributos serem iguais, entra o tempo cumulativo máximo de visita de cada subsequência $[Tc']$.

Algoritmo 3. Algoritmo para gerar recomendações

Entrada:

$s1$ Conjunto de subsequências com suporte maior que σ
qU Pedido do utilizador sobre os Itens qi que pretende visitar
qT Pedido do utilizador sobre o Tempo qt que pretende dispensar

Saída:

θ Conjunto de subsequências $< sequenceIDn\{Itemn\}[Tc'] >$ a serem recomendados

Método

```
1:  para cada sequenceID ∈ s1
2:      para cada {Item} ∈ s1
3:          para cada item {qi} ∈ qU
4:              se{Itemn} ∈ s1 = {qi} ∈ qU então
5:                  adicionar sequenceIDn a θ
6:                  adicionar {Itemn} a θ
7:              fim se
8:          fim
9:      fim
10: fim
11: para cada sequenceIDn ∈ θ
12:     para cada sequenceIDn ∈ s1
13:         se sequenceID ∈ θ = sequenceIDn ∈ s1
14:             adicionar [Tc'] a θ
15:         fim se
16:     fim
17: fim
18: para cada sequenceIDn ∈ θ
19:     para cada Tc' ∈ s1
20:         para cada qt ∈ qT
21:             se Tc' ∈ θ > qt ∈ qT então
22:                 remover linha < sequenceID{Item}Tc' > de θ
23:             fim se
24:         fim
25:     fim
26:     Ordenar θ por suporte σ, comprimento da subsequência freq e menor
        tempo cumulativo de visita [Tc']
27: fim
```

Figura 4 - Pseudocódigo para apresentar recomendações de subsequências de POIs de acordo com os requisitos do utilizador

Em anexo está o código em R para cada um destes algoritmos.

3.3 Métricas e Recomendação de sequência de POIs

Ao implementar um sistema de recomendação é preciso ter em mente duas questões: o que deve estar numa lista de recomendação e como é que sabemos que se trata de uma boa recomendação. Assim, a avaliação de um sistema de recomendação é uma etapa muito importante na conceção de um sistema de recomendação. Permite dar indicações se o modelo fornece boas ou más recomendações e, deste modo, analisar o comportamento e a

performance do sistema quando testado num cenário real.

A qualidade de um algoritmo de recomendação pode ser avaliada utilizando diversos tipos de métricas, algumas mencionadas no **subcapítulo 2.4** desta tese, que são utilizadas dependendo dos dados e do tipo de técnica de recomendação que se aplica.

Uma vez que não existem dados de *ranking* de utilizadores ou itens, optou-se por restringir as sugestões dadas ao utilizador partindo do princípio que este, ao utilizar a plataforma do MDUP, sabe com antecedência pelo menos um POI que quer visitar e o tempo que pretende gastar durante a sua visita (que pode incluir outros POIs além do requerido).

Desta maneira, o código percorre a lista de subsequências identificadas pelo algoritmo SPADE e encontra as que contêm o(s) POI(s) identificados pelo utilizador dentro do intervalo de tempo escolhido. Estas sugestões são ordenadas com as seguintes regras:

- Apresentar as subsequências com um suporte maior, ou seja, aquelas subsequências cujos POIs aparecem mais vezes nas rotas pré-definidas;
- Apresentar as subsequências que tenham um maior número de POIs para o tempo definido pelo utilizador;
- Apresentar as subsequências com o menor tempo máximo de visita.

Estas restrições permitem identificar, em princípio, a subsequência de POIs mais adequada para cada utilizador.

3.4 Tecnologias utilizadas

Há diversas ferramentas online e gratuitas para implementar sistemas de recomendação e com algoritmos disponíveis para serem utilizados sem custo adicional. Assim, selecionou-se o software open-source R-*studio,* utilizado frequentemente tanto para análises de dados como para programação. Assim, aplicou-se o R para visualizar e analisar os dados e para programar o sistema de recomendação.

Tanto a base de dados contendo as rotas do MDUP como a app que apresenta o resultado da recomendação ao utilizador é realizada pela WebLevel. Desta maneira foi necessário providenciar que o R tenha um código, neste caso para importar os dados do MySQL e, em segundo lugar, para exportar os resultados em JSON.

4. Experiência e Resultados

Esta secção apresenta os resultados da experiência. Nesta secção faz-se uma súmula do objetivo do projeto, uma análise exploratória e a preparação dos dados recolhidos. Numa segunda parte, apresentam-se os resultados das experiências efetuadas.

4.1 Objetivo

O uso de apps para a área do turismo tem vindo a aumentar de ano para ano. Tanto para definir rotas em museus, como para definir rotas em diferentes cidades em diferentes tipos de POIs. Faz sentido que além de uma visualização de diferentes rotas pré-definidas a app contenha também uma área onde é possível recomendar uma sequência de POIs que vá de encontro à necessidade de informação e interesses do utilizador/visitante. Ainda que se possam definir públicos alvo, o projeto visa tocar o público em geral. Assim, o projeto MDUP inclui uma fase dedicada à implementação de sistema de recomendação discutida nesta tese. Como foi referido, o projeto não disponibilizou dados e, portanto, utilizar-se-á uma estrutura de dados semelhante à do projeto, recolhida do artigo de Tsai et al. (2015).

Para isso selecionou-se um algoritmo capaz de identificar subsequências de POIs frequentes nas rotas. E com isso procurou-se identificar a similaridade entre o requisito do utilizador da app e as subsequências obtidas. Assumiu-se que o utilizador da app sabe pelo menos um POI que quer visitar e o tempo que pretende dispensar na visita da sequência de POIs recomendados.

Uma vez que queremos perceber quais os POIs mais frequentes em cada rota é necessário estabelecer algumas diretrizes:

1. É necessário haver um id único que identifique a rota
2. Cada POI só poderá aparecer uma vez em cada rota
3. É necessário uma variável que identifique o tempo de visita em cada POI
4. É necessário calcular o tempo cumulativo de cada POI em cada rota e assumir que inclui o tempo de percurso entre POIs

4.2 Análise exploratória dos dados

Para que o resultado final seja de alguma maneira semelhante ao implementado no MDUP preparou-se um ficheiro *Comma-Separated Values* (CSV) com os dados indicados no artigo de Tsai et al. (2015), alterando a coluna tempo para o tempo individual de visita de cada POI e não o cumulativo como está no artigo. Assim, para cada rota, subtraiu-se o valor de tempo em cada linha contígua para determinar o valor de tempo de visita. Após esta pré-preparação dos dados, procedeu-se à importação do ficheiro no software a procedeu-se à análise exploratória dos dados. A etapa tem como objetivo explorar os atributos dos dados, individualmente ou em conjunto, por forma a extrair as principais caraterísticas dos dados através de estatísticas e definir um plano de ação.

Note-se que, apesar de o código conter uma linha para remoção de dados em falta (NA's), não se espera que haja dados em falta. Se ocorrer, é provavelmente um caso acidental. A linha para remoção de NA's existe apenas como prevenção de conflitos na execução do código.

A amostra utilizada na tese consiste em 39 registos divididos por três colunas: id Rota, Tempo, em minutos, e POI, como referido no **subcapítulo 3.1** desta tese. As duas primeiras variáveis são quantitativas discretas, e os itens variáveis qualitativas nominais. As Tabelas seguintes apresentam um sumário dos dados utilizados para elaborar o sistema de recomendação.

Tabela 3 - Sumário dos dados importados em CSV no R

Indicador	**Nº de Observações**
Número de Rotas	6
Número diferente de POIs	12
Tempo mínimo de visita de um POI (minutos)	1
Tempo máximo de visita de um POI (minutos)	23
Número total de registos	39

Tabela 4 - Sequência de POIs e tempo cumulativo das 6 rotas do conjunto de dados

IDRota	Sequência {Item}[tempo cumulativo gasto na visita]
100	<{k11}[4],{k1}[5],{k2k3}[14],{k4}[20],{k7}[38],{k8}[52],{k12}[60]>
200	<{k11}[1],{k1}[20],{k3}[39],{k4}[46],{k5}[47],{k7}[50],{k12}[60]>
300	<{k11}[8],{k1}[9],{k3}[25],{k4}[37],{k5}[39],{k12}[54]>
400	<{k11}[2],{k1}[7],{k3}[17],{k4}[27],{k10}[46],{k8}[53],{k12}[60]>
500	<{k11}[1],{k1}[2],{k3}[14],{k4}[19],{k10}[40],{k12}[60]>
600	<{k11}[7],{k1}[30],{k2k3}[44],{k12}[58]>

Pelas tabelas em cima percebe-se que há seis rotas pré-definidas no sistema, sendo que o primeiro e último POI é sempre o mesmo, referente à entrada e saída do parque de diversões mencionado no artigo. Cada rota tem no máximo sete POIs, e o tempo máximo de visita do parque é de 60 minutos.

Como referido trata-se de um projeto numa fase inicial e o número de dados disponibilizados é muito reduzido. Tal acontece com os dados usados nesta tese como forma de testar o sistema de recomendação. No entanto, estes poucos dados obtidos no artigo permitirão pelo menos obter subsequências frequentes uma vez que há vários POIs que se repetem ao longo das várias rotas. A figura 5 apresenta a frequência dos diferentes POIs no conjunto de dados

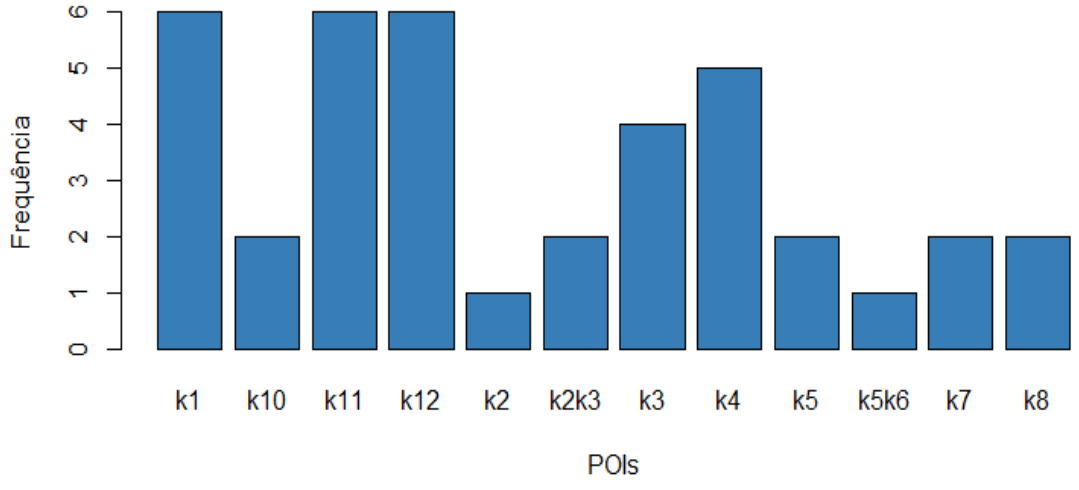


Figura 5 - Frequência de cada POI nas 6 rotas do conjunto de dados

Como esperado num conjunto de dados com itens há sempre aqueles mais ou menos visitados. Segundo a figura em cima há uma grande discrepância entre alguns dos POIs. Sendo o artigo aplicado a um parque temático é espectável que **k11** e **k12** apareçam nas 6 rotas pois correspondem à entrada e saída respetivamente. Este gráfico diz-nos já que a probabilidade de serem recomendados POIs como **k2** ou **k5k6** é bastante improvável, mas o **k1** ou **k4** têm uma maior probabilidade de aparecerem nas subsequências.

A figura seguinte apresenta os tempos associado a cada POI nas diversas rotas.

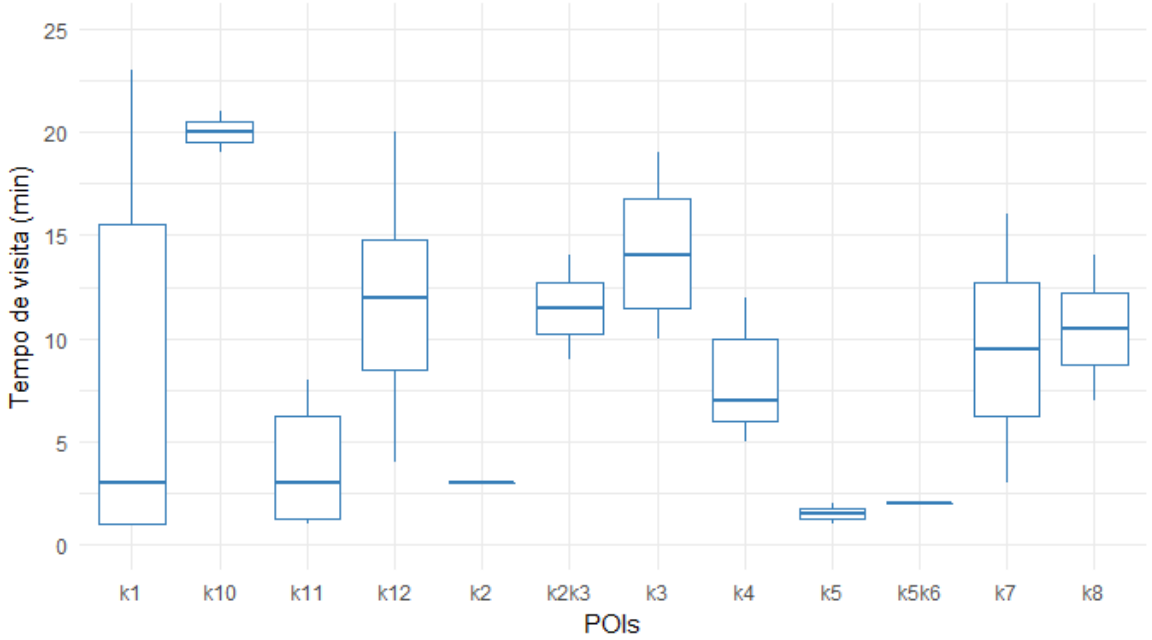


Figura 6 - Tempo de visita de cada POI nas 6 rotas do conjunto de dados

A figura 6 permite-nos perceber se a modelação do percurso face ao tempo gasto em cada POI vai afetar muito o resultado final. Ou seja, se a variação de visita de um POI é tal que implique depois uma grande diferença entre o tempo máximo e mínimo de percurso. Isso acontece com k1, sendo k1 o segundo POI que aparece em cada uma das rotas é também aquele que apresenta os valores mais díspares (entre dois e vinte e três). Quando for recomendado, este POI vai ser responsável por uma amplitude muito grande de tempo de visita de uma subsequência. Num roteiro turístico, este cenário pode repetir-se e é importante ter a noção de quanto um POI pode alterar a perceção do tempo do percurso total. Consequentemente, será importante, também, analisar a sua relevância para a geração de subsequências e reconsiderar a eliminação de POIs, tal como se fez para aqueles

que aparecem apenas uma vez.

Os dados recebidos têm algumas limitações, sendo que a mais importante é a quantidade de rotas disponibilizadas, seguida da relativa ao tempo, uma vez que não sabemos a duração do percurso entre POIs (temos apenas informação do tempo de visita em cada um). A terceira limitação diz respeito à caracterização do que é um sistema de recomendação típico. De facto, apenas há dados relativos aos POIs mas nada quanto à sua popularidade (ratings) entre diversos tipos de utilizadores, pois não há ainda nenhum utilizador registado. Perante tudo isto, é importante olhar para estes dados como um ponto de partida para colmatar alguns aspetos do *cold-start* mas repensar o sistema de recomendação, assim que haja mais dados.

4.3 Preparação dos dados

O passo da preparação dos dados é fundamental para que a recomendação ao utilizador funcione da maneira pretendida, tendo em vista obter as subsequências mais frequentes.

Assim que os dados foram descarregados no R, optou-se por criar uma linha de código que remove as linhas com NA's, pois um valor em falta numa base de dados com rotas deve-se, muito provavelmente, a um erro ou uma falha no sistema. Todas as rotas têm que ter um RotaID, um POI e um tempo de visita associado.

De seguida optou-se por identificar os POIs mais frequentes (ver figura 5), que aparecem pelo menos duas vezes nas rotas, para criar subsequências dos POIs que são, de uma maneira indireta, mais populares. Para evitar que alguns POIs possam não ser visitados, sugere-se a apresentação de um mapa com a rota e os POIs mais próximos para que seja o utilizador a procurá-los, caso seja do seu interesse.

Criou-se uma coluna com o tempo de visita cumulativo para cada RotaID. Esta opção permite-nos, primeiro, assumir que o tempo de percurso entre POIs não é relevante face ao tempo gasto em cada visita; segundo, preparar os dados em forma de transação (ver tabela 4 com o tempo total de cada rota).

De maneira a que os dados sejam processados pelo algoritmo SPADE (função cSpade no R) é necessário transformá-los numa transação. Para isso, ordenou-se o ID das rotas e o tempo de visita cumulativo de maneira descendente. Criou-se um novo ficheiro, requisito do algoritmo SPADE no R, e fez-se novamente o carregamento do ficheiro no R.

4.4 Modelação

A fase de modelação tem o propósito de encontrar a solução mais eficiente para gerar as subsequências e as recomendações, com base nessas subsequências, ao utilizador.

Para obter as subsequências mais frequentes recorreu-se ao algoritmo SPADE no R. É importante selecionar o suporte mínimo adequado ao conjunto de dados que se tem e ao objetivo do projeto. Um suporte mínimo muito alto pode resultar na não devolução de dados ou na não seleção de POIs que até poderiam ser interessantes recomendar. Com um conjunto de dados muito grande, um suporte mínimo muito baixo pode ser impraticável, devolvendo demasiados resultados, sendo difícil confirmar qual a recomendação mais adequada ao utilizador. Acrescido a esse problema há o próprio processamento dos dados, e, consequentemente, uma resposta muito demorada na recomendação ao utilizador. Por estes motivos, é importante encontrar o equilíbrio ideal.

A figura 7 apresenta o comprimento de cada subsequência e a sua frequência para diferentes valores de suporte. Esta representação pode ajudar na seleção do suporte mais indicado para o conjunto de dados que temos. Por exemplo, se estivermos a contar em recomendar subsequências com pelo menos seis POIs, esta representação ajuda-nos a perceber qual o suporte mínimo que teríamos de definir.

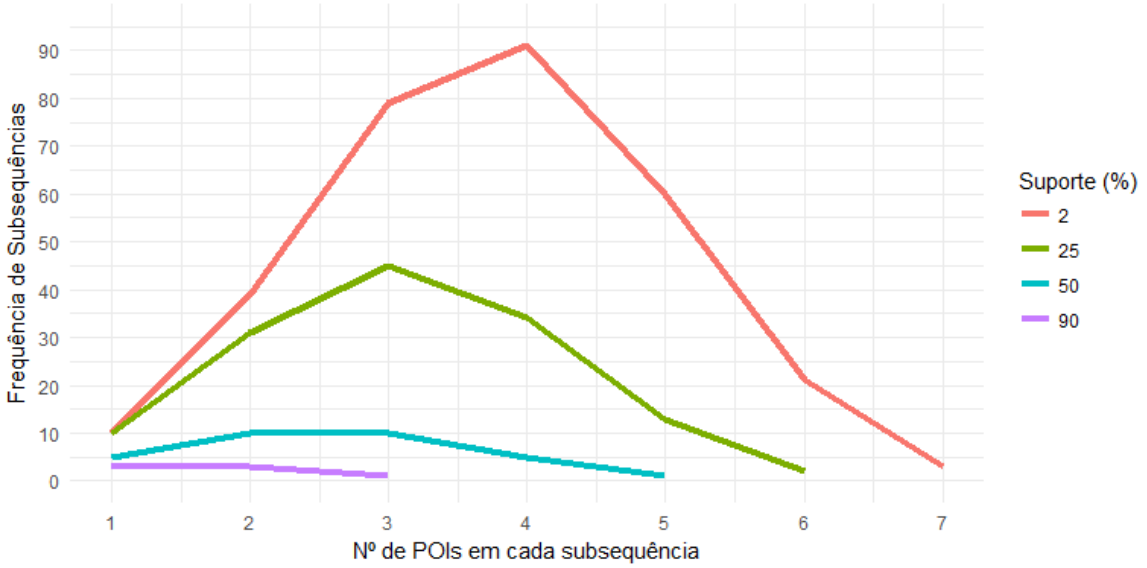


Figura 7 - Frequência e número de POIs em cada subsequências e obtidas pelo SPADE para diferentes valores de suporte.

Tal como se pode observar na figura 7 um suporte muito elevado não devolve muitas subsequências, nem subsequências com um número relevante de POIs para visitar. Definindo um suporte muito baixo, o número de subsequências devolvidas aumenta

drasticamente. Ou seja, para um suporte de 90% temos um total de sete subsequências para recomendar, em que o tamanho máximo da subsequência é de três itens, e há três subsequências com apenas um item, enquanto que com um suporte de 2% é possível obter 303 subsequências, 10 das quais com apenas um item. A seleção deste suporte já permite recomendar três subsequências de sete itens. Esta análise é relevante na medida em que para um conjunto de dados com um maior volume pode acontecer que 90% também devolva um conjunto suficiente de subsequências.

Uma vez que o número de dados é pequeno, optou-se por definir o suporte mínimo de 2%, significando que mesmo no caso das subsequências que apenas aparecem 2% das vezes vão ser devolvidas na tabela final. É de notar que das subsequências obtidas é importante eliminar as subsequências redundantes, ou seja, uma subsequência prossupõe ter pelo menos dois POIs. Assim, selecionaram-se as 10 subsequências mais frequentes, para um suporte mínimo de 2%, com pelo menos dois itens que se apresentam na figura 8.

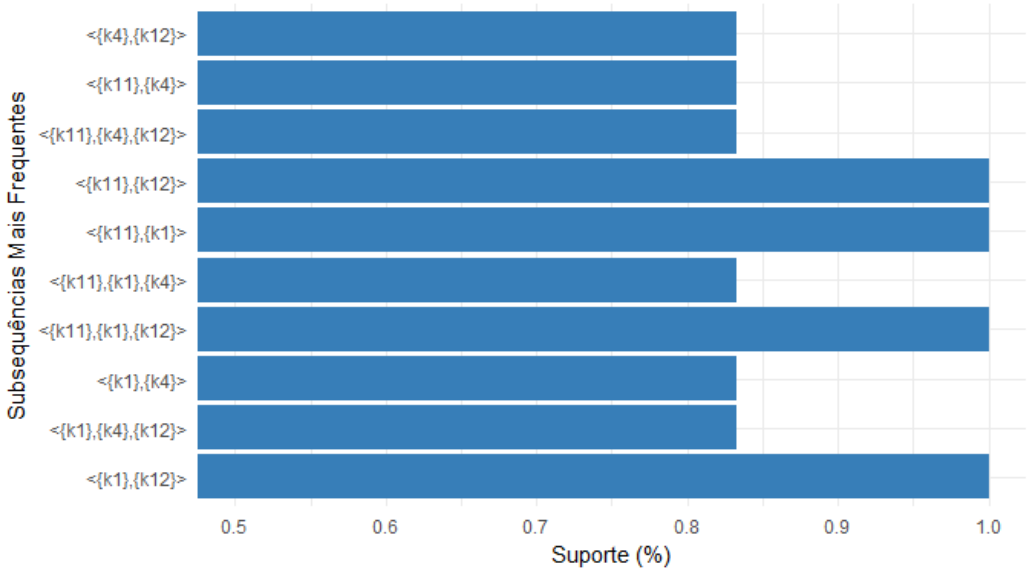


Figura 8 - Lista das 10 subsequências mais frequentes obtidas a partir do algoritmo SPADE, para um suporte mínimo de 2%

Por exemplo, a primeira linha da figura 8 indica que numa rota o POI **k4** é normalmente seguido do POI **k12** numa visita. Talvez por proximidade ou por serem do mesmo género. Essa tendência de um visitante ir do POI **k4** para o POI **k12** é frequente para cerca de 80% das rotas na base de dados. E, como esperado, devido à sua alta frequência nas diferentes rotas (ver figura 5), o POI **k4** aparece em várias subsequências com suporte elevado.

Aliás, se procurarmos as regras de associação das subsequências apresentadas na figura 8, para o suporte mínimo definido de 2%, obtemos os seguintes resultados para as primeiras

dez subsequências:

Tabela 5 - Regras de associação para um suporte mínimo de 2%

Regra de Associação	Suporte (%)	Confiança (%)
<{k1}> => <{k12}>	100	100
<{k11}> => <{k12}>	100	100
<{k11},{k1}> => <{k12}>	100	100
<{k11}> => <{k1}>	100	100
<{k1}> => <{k4}>	83.3	83.3
<{k11}> => <{k4}>	83.3	83.3
<{k11},{k1}> => <{k4}>	83.3	83.3
<{k4}> => <{k12}>	83.3	100
<{k11},{k4}> => <{k12}>	83.3	100
<{k1},{k4}> => <{k12}>	83.3	100

Na tabela 5 consegue-se perceber como o algoritmo configura as regras para considerar uma subsequência válida. Por exemplo, a última regra **<{k1},{k4}> => <{k12}>** significa que se um utilizador visita **{k1},{k4}** é bastante provável que visite o POI **{k12}**. O valor referente ao suporte significa que 83% das subsequências totais contêm **<{k1},{k4},{k12}>** como subsequência. Por outro lado, o valor da confiança significa que 100% das vezes que a subsequência **<{k1},{k4}>** acontece, **<{k12}>** acontece também, o que é lógico pois este é o POI referente à saída do parque.

O algoritmo utilizado para gerar subsequências permite-nos ter uma noção do tempo que se irá gastar no percurso de cada subsequência. A tabela 6 apresenta um sumário para as subsequências mencionadas na figura 8 e tabela 5. Os resultados estão ordenados pelo comprimento da subsequência, ou seja, de acordo com o número de POIs que existem na subsequência.

Tabela 6 - Tempo cumulativo e frequência das subsequências

Subsequência	Tempo mínimo percurso (min)	Tempo máximo percurso (min)	Nº de POIs na subsequência
<{k11},{k1},{k12}>	6	51	3
<{k11},{k1},{k4}>	7	43	3
<{k11},{k4},{k12}>	10	40	3
<{k1},{k4},{k12}>	10	55	3
<{k1},{k12}>	5	43	2
<{k11},{k12}>	5	28	2
<{k11},{k1}>	2	31	2
<{k1},{k4}>	6	35	2
<{k11},{k4}>	6	20	2
<{k4},{k12}>	9	32	2

Como esperado, as subsequências com maior número de POIs implicam um maior tempo máximo de visita. Também é de referir que, graças à amplitude de valores para o tempo gasto a visitar **k1** (ver figura 6), na prática o tempo gasto na subsequência **{k1, k4}** será sobretudo em **k4**. Continua a verificar-se uma grande amplitude de valores entre o tempo mínimo e máximo de visita, que tem a sua lógica porque são sobretudo resultados referentes aos primeiros POIs ou ao último do parque de diversões em estudo no artigo.

De seguida foi criado um código para avaliar a semelhança entre o pedido do utilizador e as subsequências obtidas. O objetivo é sugerir a subsequência de POIs que vai de encontro aos POIs selecionados pelo utilizador dentro do tempo estipulado pelo mesmo. A sugestão é dada de acordo com o suporte e a frequência da subsequência, ou seja, o número de POIs que existem na subsequência e no caso de os valores serem idênticos considera-se o fator tempo. Por exemplo, considerando que foi usado um suporte de 2% na determinação de subsequências mais frequentes, se um utilizador da app tiver como preferência no seu percurso os itens **{k1, k3}** e um tempo máximo de 90 minutos para andar a visitar o local as recomendações sugeridas seriam as seguintes:

Tabela 7 - Exemplo de recomendação

Recomendação	Subsequência de POIs recomendados	Suporte (%)	Tempo de visita expectável
# 1	{k11, k1, k3}	67	12 a 50 minutos
# 2	{k1, k3, k5}	33	12 a 44 minutos
# 3	{k1, k3}	67	11 a 42 minutos

Como referido, a lista ordenada de sugestões tem como base, primeiro o suporte da subsequência, e em segundo lugar a quantidade de POIs existente na subsequência. Ordenaram-se os dados por tempo máximo de visita no caso de empate entre suporte e frequência. Olhando para a tabela acima, à primeira vista a recomendação **#2** deveria ser aquela que surgiria em primeiro lugar. Se modificarmos o suporte para 50%, a recomendação devolve-nos apenas a primeira e a segunda recomendação pois o suporte da subsequência em **#2** é de apenas 33%.

4.5 Avaliação

Na fase de avaliação pretende-se verificar se o modelo sugerido vai de encontro ao objetivo do projeto. No entanto, devido à falta de dados de *ranking* de utilizadores tal passo não é possível fazer em toda a plenitude. Assim, considera-se que se fazem boas recomendações sempre que a sugestão vai de encontro ao que o utilizador pediu. Também se assume que o utilizador preferirá ver uma subsequência com um suporte mais elevado, intrinsecamente mais popular, e que quererá ver o maior número de POIs em sequência além daqueles que requisitou dentro do tempo que estipulou previamente.

A recolha do histórico e de *rankings* dos utilizadores que usarem este sistema de recomendação poderão ser depois ser utilizados para verificar quão bom é este sistema de recomendação. Esta recolha de dados terá que estar de acordo com o Regulamento Geral de Proteção de Dados (RGPD) que entrou em vigor em 25 de Maio de 2018.

4.6 Plano de Utilização

A fase de implementação do sistema de recomendação estará a cargo da UP Digital e da WebLevel através de uma app. O código desta tese fará parte do sistema *backend* que fará a sugestão ao utilizador de POIs sobre o que visitar através das subsequências obtidas das rotas da base de dados. A manutenção e a determinação da sua frequência também estará a cargo destas organizações. A partir do momento em que se carregam os dados na base de dados e no R não é necessário repetir o processo para gerar subsequências diariamente e perder tempo no processamento dos dados, já que é pouco provável que haja novas rotas todos os dias.

O único requisito por parte da WebLevel foi a criação de uma linha de código para exportar os resultados em JSON. Este ficheiro inclui a sequência de POIs, o tempo máximo e mínimo espectável de visita da subsequência, e também, o tempo associado a cada POI individual de cada subsequência para a eventualidade de terem que trabalhar com esses dados ainda no *backend* antes de serem submetidos na app.

5. Conclusões e perspetivas de desenvolvimento

O turismo é uma parte significativa do mundo atual e as apps de recomendação são cada vez mais utilizadas para viagens de curta ou longa duração. Estas viagens têm normalmente um tempo definido para visita e os interesses podem variar de pessoa para pessoa, ou mesmo de grupo para grupo (uma escola, *team building*, etc.). O projeto do MDUP vai de encontro a esta crescente procura do património científico, museológico, arquitetónico, etc. do nosso país, e, em parceria com a WebLevel, entre outras empresas, entrou-se na fase de criação de uma app em que se apresentam as diferentes rotas recolhidas por outros projetos do MDUP, como #IWASHERE e #GPSEngenharia.

Esta tese tem como objetivo a criação de um sistema de recomendação para os utilizadores dessa app, de maneira a sugerir o que de melhor esta cidade tem apara oferecer de acordo com os seus gostos e tempo disponível.

Normalmente os sistemas de recomendação nesta área são apoiados em *rankings* e opiniões de utilizadores que já tenham visitado os POIs para sugerir possíveis destinos a novos utilizadores. Devido ao estado embrionário do projeto os dados existentes são apenas sobre o património ou itens de outra natureza e o tempo típico de visita de cada POI sob a forma de rotas. Assim, para identificar os POIs mais adequados dentro do tempo disponível do utilizador esta tese propõe um sistema de recomendação de subsequências de POIs baseado no algoritmo de mineração de dados em sequência, SPADE. O algoritmo permite recomendar uma sequência de POIs, que se podem considerar mais populares e importantes, ou seja, identifica e sugere aqueles que aparecem mais vezes nas rotas elaboradas pelo projeto MDUP, dentro do limite de tempo estipulado pelo utilizador. Com esta solução estamos a recomendar uma subsequência de POIs que aparecem com mais frequência – suporte elevado – indiretamente relacionados com a popularidade de um POI e recomendamos uma subsequência que tenha mais variedade de POIs, mas sempre tendo em consideração o tempo estimado de visita do utilizador.

Uma vez que o projeto está numa fase inicial, e os dados tem características sobretudo dos POIs decidiu-se não colmatar este *cold-start* com dados de utilizadores baseados em demografia. Ou seja, considerando utilizadores tipo, idade, sexo, formação escolar, para cada rota anexada ao projeto. O número de rotas é bastante limitado e a criação de grupos tipo poderia causar alguns problemas na execução do sistema de recomendação pois continua a não haver uma avaliação histórica dos POIs. Assim, optou-se por adaptar o modelo de Tsai et al. (2015) por se aproximar dos dados existentes do projeto.

Devido à indisponibilidade dos dados do projeto MDUP para elaboração desta tese testou-se o sistema de recomendação com os dados apresentados no artigo. Os dados contêm

apenas 39 registos distribuídos por seis rotas. Os dados utilizados nesta tese tiveram como objetivo seguir uma estrutura semelhante ao projeto MDUP, ter uma melhor perceção do funcionamento do sistema de recomendação, por não ser um volume de dados muito elevado, e avaliar como se adaptará aos poucos dados que o MDUP tem neste momento.

Os atributos dos dados utilizados consistem no identificador da rota, no identificador do POI e no tempo associado à visita desse POI. Considerou-se que o tempo de visita total não será muito diferente do tempo de visita do percurso mais visitado. Para obter um grande número de subsequências e com um maior comprimento definiu-se um suporte mínimo de 2%. Como resultado obteve-se 303 subsequências cuja subsequência de maior comprimento tem sete POIs com uma frequência de três.

Para se fazer a recomendação requere-se que o utilizador indique pelo menos um POI que queira visitar e o tempo que deseja passar nesse percurso. Elaborou-se assim um código que analisa a similaridade entre as preferências de visita do utilizador, do POI e tempo, com o resultado das subsequências devolvidas pelo algoritmo SPADE.

A experiência apresentou um sistema de recomendação robusto cuja avaliação da eficácia é feita com a recomendação baseada no suporte da subsequência, ou seja, serão apresentadas primeiro as subsequências indiretamente mais populares, na frequência de POIs dessa subsequência, ou seja, entre uma subsequência com o mesmo suporte, é sugerida a subsequência com um maior número de POIs, e por último, no caso do suporte e frequência serem o mesmo é sugerida a subsequência que tenha um menor valor do tempo máximo de visita.

A solução apresentada nesta tese tenta colmatar algumas limitações existentes, o número reduzido de dados disponível, a falta de dados sobre utilizadores, e mesmo a falta dos dados do próprio projeto para testar o sistema de recomendação. Apesar de ser uma solução adaptada ao estado embrionário do projeto há problemas de *cold-start*. O algoritmo testado seleciona os POIs mais frequentes para traçar uma rota para o utilizador, a exclusão de POIs menos frequentes, ou que foram apenas recentemente adicionados ao sistema pode implicar que nunca sejam sugeridos. A sugestão para este problema é a apresentação da rota ao utilizador na forma de um mapa com os POIs nas redondezas, frequentes e não frequentes, para que assim o utilizador possa optar por sair do percurso sugerido e visitá-los. Outra sugestão adicional a este problema é o registo do histórico das visitas efetuadas para que assim se formem outras rotas, com outros POIs e tempos de visita, mas sempre com a atenção ao RGPD.

Perspetivas Futuras

Este sistema de recomendação é o primeiro passo para um sistema de recomendação mais robusto e eficaz. A inclusão de uma definição de perfil de utilizador na app poderá permitir que o sistema comece a recomendar POIs específicos adaptados a grupos característicos de utilizadores. A título de exemplo: um utilizador que goste sobretudo de arte, o sistema de recomendação tem de ser capaz de sugerir subsequências que incluam sobretudo POIs ligados à arte. A possibilidade de gravar o histórico do percurso de cada utilizador pode também permitir a criação de novas rotas, grupos de interesse, e a até uma modelação mais precisa sobre o tempo de visita dessa rota. Também será interessante disponibilizar a possibilidade de avaliar a visita a determinado POI de maneira a tornar um sistema de recomendação baseado em filtragem colaborativa ou de conteúdo.

Para incorporar o tempo de percurso entre POIs neste sistema de recomendação há algumas alternativas possíveis. Por exemplo, a WebLevel após receber a lista de recomendação poderia incorporar as subsequências num mapa e calcular a distância entre POIS. A distância de percurso de cada subsequência seria somada ao tempo total de visita que fora devolvido pelo sistema de recomendação. Ou, por exemplo, usar algoritmos de *clustering* (como *K-means*) para criar *clusters* de POIs de acordo com o tempo de percurso. Estes dados ajudarão a filtrar as recomendações, permitindo remover um POI da subsequência ou mesmo uma das subsequências para tornar uma recomendação eficaz.

Há que estar atento à terceira geração de sistemas de recomendação que usarão a web 3.0, que inclui dados relacionados com contexto, que recolhem dados através de sensores (temperatura, hábitos alimentares, etc.) que permitirão uma recomendação mais adequada e direcionada a cada utilizador. São sistemas de recomendação híbridos, considerados altamente eficientes para recomendar o percurso ideal a um utilizador.

Referências bibliográficas

Aggarwal, Charu C. 2015. Data Mining - The Textbook. Springer International Publishing Switzerland, DOI 10.1007/978-3-319-14142-8.

Aggarwal, Charu C. 2016. Recommender Systems - The Textbook. Springer International Publishing Switzerland. DOI: 10.1007/978-3-319-29659-3.

Batet, Montserrat, Antonio Moreno, David Sánchez, David Isern e Aïda Valls. 2012. "Turist@: Agent-based personalised recommendation of tourist activities". Expert Systems with Applications, 39 : 7319–7329. DOI: 10.1016/j.eswa.2012.01.086

Bellogín, Alejandro e Pablo Sánchez. 2017. “Collaborative filtering based on subsequence matching: A new approach”. Information Sciences, 418–419 : 432–446. DOI: 10.1016/j.ins.2017.08.016.

Benouaret, Idir e Dominique Lenne. 2015."Personalizing the Museum Experience through Context-aware Recommendations". IEEE International Conference on Systems, Man, and Cybernetics. DOI: 10.1109/SMC.2015.139

Binucci, Carla, Felice De Luca, Emilio Di Giacomo, Giuseppe Liotta e Fabrizio Montecchiani. 2017. "Designing the Content Analyzer of a Travel Recommender System". Expert Systems With Applications, 87 : 199–208 DOI: 10.1016/j.eswa.2017.06.028.

Bobadilla, Jesús, Fernando Ortega, Antonio Hernando e Abraham Gutiérrez. 2013. “Recommender systems survey”. Knowledge-Based Systems, 46 : 109–132. DOI: 10.1016/j.knosys.2013.03.012.

Borràs, Joan, Antonio Moreno e Aida Valls. 2014. "Intelligent tourism recommender systems: A survey". Expert Systems with Applications, 41 : 7370–7389. DOI: 10.1016/j.eswa.2014.06.007

Burke, Robin. 2002. “Hybrid Recommender Systems: Survey and experiments”. User Modeling and User-Adapted Interaction, 12(4) : 331–370. DOI: 10.1023/A:1021240730564

Campos, Pedro G., Fernando Díez e Iván Cantador. 2014. "Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols". User Model User-Adap Inter, 24 : 67–119 DOI: 10.1007/s11257-012-9136-x.

Cai, Guochen, Kyungmi Lee e Ickjai Lee. 2018. "Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos". Expert Systems With Applications, 94 : 32–40. DOI: 10.1016/j.eswa.2017.10.049

Hernando, Antonio, Jesús Bobadilla, Fernando Ortega e Abraham Gutiérrez. 2017. "A probabilistic model for recommending to new cold-start non-registered users". Information Sciences, 376 : 216–232. DOI: 10.1016/j.ins.2016.10.009

He, Ruining e Julian McAuley. 2016. "Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation". IEEE 16th International Conference on Data Mining. DOI: 10.1109/ICDM.2016.0030

Kurataa, Yohei e Tatsunori Harab. 2014. "CT-Planner4: Toward a More User-Friendly Interactive Day-Tour Planner". Information and communication technologies in tourism 73–86. DOI: 10.1007/978-3-319-03973-2_6

Lika, Blerina, Kostas Kolomvatsos e Stathes Hadjiefthymiades. 2014. "Facing the cold start problem in recommender systems". Expert Systems with Applications 41: 2065–2073. DOI: 10.1016/j.eswa.2013.09.005

Park, Deuk Hee, Hyea Kyeong Kim, Il Young Choi e Jae Kyeong Kim. 2012. “A literature review and classification of recommender systems research”. Expert Systems with Applications, 39 : 10059–10072. DOI: 10.1016/j.eswa.2012.02.038.

Peng, Furong, Jianfeng Lu, Yongli Wang, Richard Yi-Da Xu, Chao Maa e Jingyu Yang. 2016. "N-dimensional Markovrandom field prior for cold-startrecommendation". Neurocomputing, 191 :187–199. DOI: 10.1016/j.neucom.2015.12.099

Pinto, Maria M., Susana Medina, Rodolfo Matos e Paulo Fontes. 2016. “U.OPENLab methodology: A Conceptual Model and Flowchart for the Dynamic Co-Production and (Re)use of Digital Contents”. Paper presented at the ICERI2016 Conference 14th-16th, Seville, Spain. DOI: 10.21125/iceri.2016.2140

Ricci, Francesco, Lior Rokach, Bracha Shapira e Paul B. Kantor. 2011. Recommender Systems – Handbook. Springer Science+Business Media, LLC,. DOI 10.1007/978-0-387-85820-3.

Ruotsalo, Tuukka, Krister Haavh, Antony Stoyanov, Sylvain Rochef, Elena Fani, Romina Deliai, Eetu Mäkeläa, Tomi Kauppinena e Eero Hyvönena. 2013. "SMARTMUSEUM: A mobile recommender system for the Web of Data". Web Semantics: Science, Services and Agents on the World Wide Web 20 : 50–67. DOI: 10.1016/j.websem.2013.03.001

Semeraro, Giovanni, Pasquale Lops, Marco de Gemmis, Cataldo Musto e Fedelucio Narducci. 2012. "A Folksonomy-Based Recommender System for Personalized Access to Digital Artworks". ACM Journal on Computing and Cultural Heritage, 5(3) : 11-22. DOI: 10.1145/2362402.2362405

Solima, Ludovico, Maria Rosaria Della Peruta e Vincenzo Maggioni. 2016. “Managing adaptive orientation systems for museum visitors from an IoT perspective”. Business Process Management Journal, 22(2) : 285-304. DOI: 10.1108/BPMJ-08-2015-0115.

Tran, Truyen, Dinh Phung e Svetha Venkatesh. 2016. “Collaborative filtering via sparse Markov random fields”. Information Sciences, 369 : 221–237 DOI: 10.1016/j.ins.2016.06.027.

Tsai, Chieh-Yuan e Bo-Han Lai. 2015. "A Location-Item-Time sequential pattern mining algorithm for route recommendation". Knowledge-Based Systems, 73 : 97–110. DOI: 10.1016/j.knosys.2014.09.012.

Zaki, Mohammed. 2001. “SPADE An Efficient Algorithm for Mining Frequent Sequences”. Machine Learning, 42(1-2) : 31–60. DOI: 10.1023/A:1007652502315

Recursos online

#GPSEngenharia. https://goo.gl/p1YqYo (consultado em jan 2018)

#IWASHERED https://goo.gl/j9Ycz6 (consultado em jan 2018).

Elsevier B.V. “Science Direct”, http://www.sciencedirect.com/(consultado em jan 2018)

Elsevier B.V. “Scopus”, https://www.scopus.com (consultado em jan 2018)

ICOM – International Council of Museums Portugal. “Definição: Museu.” ICOM - Portugal, http://icom-portugal.org/2015/03/19/definicao-museu/ (consultado em jan 2018)

WebLevel – Tecnologias de Informação, https://www.weblevel.pt/ (consultado em jun 2018)

Anexos

Este anexo apresenta o código escrito em R correspondentes aos Algoritmos 1, 2 e 3 desta tese (Figuras 2, 3 e 4). Apresenta também o código sob a forma de comentário para se importar os ficheiros de uma base de dados SQL e o código para exportação de um ficheiro com formato JSON.

```
# Packages utilizadas
library("plyr")
library("dplyr")
library("arules")
library("Matrix")
library("arulesSequences")
library("tidyr")
library("readr")
library("jsonlite")

##Código para importação do MySQL
#library(RODBC)
#con <- dbConnect(odbc::odbc(),
# dbname='myDB',
# host='host.com',
# port=8888,
# user='myusername',
# password='123456')
#data <- dbSendQuery(con, "SELECT * FROM myTable")
#dbDisconnect(con)

##Algoritmo 1
#Carregar o ficheiro CSV, com RotaID, Tempo e Item
rotas <- read.csv("Rotas - Copy (2).csv", sep=";", header = TRUE)
#Remover os NA's
rotas <- na.omit(rotas)
#Adicionar uma coluna com a soma cumulativa do tempo
rotas <- rotas %>% dplyr::group_by(RotaID) %>% dplyr::mutate(TempoCum =
cumsum(Tempo))
rotas <- as.data.frame(rotas)
#Contar a frequência e selecionar os itens que aparecem mais que 1 vez
Sup_count = plyr::count(rotas, c("Item"))
SupCount_freq <- filter(Sup_count, freq>1)
# Agregar a informação do data.frame 'SupCount_freq' ao 'rotas'
SeqjoinRotas <- dplyr::full_join(SupCount_freq, rotas, by = "Item")
#Remover os NAs, ordenar por RotaID e TempoCum, e construir um data frame só com 3
colunas
NovaRota <- SeqjoinRotas %>%
        na.omit %>%
        arrange(RotaID, TempoCum) %>%
        select(RotaID, TempoCum, Item)
#Infelizmente tem que se criar um novo ficheiro para contruir a transação e usar o Spade
write.table(NovaRota, "transacao.txt", sep=" ", row.names = FALSE, col.names = FALSE,
quote = FALSE)
Transacao <- read_baskets("transacao.txt", info = c("sequenceID","eventID"))
```

```
##Algoritmo 2
#Resultado do algoritmo Spade com o min support 2%
s1 <- cspade(Transacao, parameter = list(support = 0.02), control = list(verbose =TRUE))
s1.df <- as(s1, "data.frame")
#Ordenar os resultados com do maior support ao mais pequeno
s1.df.1 <- s1.df %>% dplyr::arrange(desc(support))
#Adicionar uma coluna com num 1:n linhas, que vai ser o nosso ID de cada subsequência
e remover a coluna support
s1.df.2 <- cbind("ID" = 1:nrow(s1.df.1), s1.df.1)
s1.df.2 <- s1.df.2[,-c(3)]
##Passos para contar o comprimento da subsequência, ou seja, quantos itens tem cada
subsequência
#Para cada ID contar quantos itens tem cada subsequência
#Remover as vírgulas dos itens
dt <- s1.df.2 %>% separate_rows(sequence)
#Remover filas com a subsequência em branco
dt2 <- dt[!(dt$sequence == ""), ]
#Renomear a coluna 'sequence' para 'Item'
names(dt2)[2]<-"Item"
#Contar a frequência de ID, que é o mesmo que número de itens em cada subsequência
FreqPOI = plyr::count(dt2, c("ID"))
##Passos para determinar o tempo de visita de cada subsequência
#Alterar a coluna 'TempoCum' para 'Tempo'
NovaRota.dt <- SeqjoinRotas %>%
        na.omit %>%
        arrange(RotaID) %>%
        select(RotaID, Tempo, Item)
#Traspor o data frame para permitir calcular o tempo de cada item
NovaRota.df <- spread(NovaRota.dt,
RotaID,
Tempo)
#Adicionar uma coluna com o tempo máximo e o mínimo que um POI teve como visita
NovaRota.df $min <- apply(NovaRota.df[,2:7], 1, min, na.rm = TRUE)
NovaRota.df $max <- apply(NovaRota.df[,2:7], 1, max, na.rm = TRUE)
#Remover a coluna RotaID
NovaRota.df <- NovaRota.df[,-c(2:7)]
#Juntar a tabela 'NovaRota.df ' com o tempo max e min com 'dt2'.
TabelaToda <- dplyr::full_join(dt2, NovaRota.df, by = "Item")
#Calcular o tempo total de uma visita a uma subsequência, considerando ID e min e max
TempoTotal.min <- cbind(aggregate(min~ID, sum, data=TabelaToda))
TempoTotal.max <- cbind(aggregate(max~ID, sum, data=TabelaToda))
#Agregar os 3 data frames (tempo min e max visita, e a frequência dos POIs numa
subsequência)
TabelaTempoFreq <- Reduce(function(x, y) merge(x, y, all=TRUE), list(TempoTotal.min,
TempoTotal.max, FreqPOI))
```

```
##Algoritmo 3
#Função para encontrar subsequencias
find.sequences <- function(TabelaToda, user)
{
      TabelaToda$items <- as.character(TabelaToda$Item)
      ref <- c()
      if (length(user) > 0)
      {
      i<-1
            while (i <= nrow(TabelaToda))
            {
                  j <- TabelaToda[i, "ID"]
                  k <- j
                  l <- 1
                  while (j == k && l <= length(user))
                  {
                      if (user[l] == TabelaToda[i, "Item"])
                      {
                          l <- l+1
                          if (l > length(user))
                                ref <- c(ref, j)
                      }
                      i <- i+1
                      if (i > nrow(TabelaToda))
                          k <- max(TabelaToda$ID)+1
                      else
                          k <- TabelaToda[i, "ID"]
                    }
            }
      }
      return(ref)
}
#Pesquisa do utilizador para os item(ns)
user <- as.character(c("k1", "k3"))
recomitem <- find.sequences(TabelaToda, user)
#Pesquisa do utilizador para o tempo, filtrando os resultados do item ordenado pela
frequência de itens numa subsequência
user.t <- 50
recomTempo <- dplyr::filter(TabelaTempoFreq, max <= user.t) %>%
          filter(ID %in% recomitem) %>%
          arrange(desc(freq))
#Encontrar os POIs e tempos associados ao ID e item
recomendacao <- full_join(TabelaToda, recomTempo, by = "ID") %>%
          na.omit %>%
          arrange(max.y) %>%
          arrange(ID) %>%
          arrange(desc(freq))

#Enviar o output em JSON
conversaoJSON <- toJSON(recomendacao, pretty=TRUE)
#Criar um ficheiro JSON - write_json(df, path\\NOMEFICHEIRO)
write_json(conversaoJSON, "C:\\Users\\recomendacao")
```